Watch n' Check: Towards a Social Media Monitoring Tool to Assist Fact-Checking Experts

ASSUNTA CERONE

B.Eng (Hons)

POLITECNICO
DI TORINO

Supervisor: Daniele Apiletti
Associate Supervisor: Damiano Spina
Associate Supervisor: Elham Naghizade

A thesis submitted in fulfilment of
the requirements for the degree of
Master's Thesis

School of Computer Engineering
Faculty of Engineering
Politecnico di Torino
Italy

16 July 2020

Abstract

The era of social media has jeopardized the authenticity of information and the spread of fake news has been a growing phenomenon difficult to control and potentially leading to unexpected and undesired outcomes. Fact-Checking movements and methods have proved to be a fundamental process for revitalizing news broadcasting and consumption, accessing the truthfulness of claims, and giving citizens information to make more conscious political choices. Many automated systems for claims detection and veracity verification have already been developed over the years. These tools often provide valuable results, but they appear as a black box that does not provide transparency on how the output is been elaborated. In this thesis, we present an ongoing work to develop Watch n' Check, in collaboration with ABC RMIT Fact Check, an Australian Fact-Checking organization. Such a tool is the result of our research on finding the most cost-effective way to integrate data analysis in the fact-checkers daily work, engaging them in the development and giving them a transparent control of the the outcome of the analysis. It allows experts access to a large number of information streams published in real-time on Twitter, allowing efficient identification of trending topics and monitoring the changes in the news propagation dynamics, improving the impartiality in the process of targeting the statements/news to be checked.

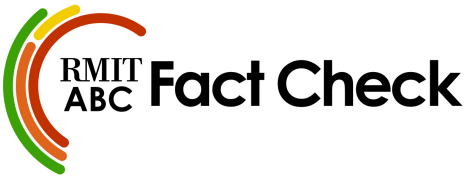
Acknowledgements

Le parole sono un mezzo di comunicazione limitante e riducono la potenza dei nostri sentimenti, razionalizzando ciò che per definizione è irrazionale.

Grazie alla mia famiglia di Muro Lucano.
Grazie alla mia famiglia di Torino.
Grazie alla mia famiglia di Gothenburg.
Grazie alla mia famiglia di Alcover.
Grazie alla mia famiglia di Melbourne.
Grazie al Politecnico di Torino per l'addestramento.
Grazie alla regione Piemonte per avermi dato la possibilità di studiare e viaggiare.
Grazie a me stessa per aver trasformato la mia *Growth Zone* nella mia *Comfort Zone*.

Stay hungry, stay foolish.

About This Work



This work has been completed in collaboration with RMIT Research Centre for Information Discovery and Data Analytics (RMIT CIDDA) and RMIT ABC Fact Check, while visiting RMIT University (Melbourne, Australia). The visit was partially supported by Politecnico di Torino (Mobilità Outgoing).

The following publication has resulted from this work:

> **Assunta Cerone**, Elham Naghizade, Falk Scholer, Devi Mallal, Russell Skelton and Damiano Spina. 'Watch 'n' Check: Towards a Social Media Monitoring Tool to Assist Fact-Checking Experts'. In: *2020 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Special Session: Fake News, Bots and Trolls*. 2020.

Contents

List of Figures

CHAPTER 1

Introduction

Social media has become a part of the fabric of our daily lives. Facebook, Twitter, YouTube and many other social networking sites allow users to share and interact with online content and to connect with like-minded people. Rapid dissemination, amplification of content, and the ability to lead informal conversations are the strengths that make it a powerful tool in many professional contexts. [41] In the same time, the authenticity of information has become a longstanding issue and the spread of fake news has been a growing phenomenon difficult to control, and potentially leading to unexpected and undesired outcome. A clear example is the new avalanche of misinformation that has accompanied the COVID-19 pandemic and has driven people in dangerous behaviours. In this scenario, the need of Fact Checking organizations and methods has proved to be a fundamental process for revitalizing news broadcasting and consumption, accessing the truthfulness of claims made by figures of influence. These movements arose in the US in the early 2000s, though an increasing number of fact-checking outlets exist now across different countries, in different organisational forms, and different self-identified orientations, but globally follow the same missions: assessing the truth of political claims and giving citizens information to make political choices. Fact checking outlets financially rely mainly on media industries and charitable foundations, thus face significant financial challenges in order to support their goals. Therefore, the integration of software and data analytics tools in these organizations could potentially speed up their daily work. Building a method to automatically check the truthfulness of news has seen an increasing number of researches in a variety of disciplines including natural language processing, machine learning, knowledge representation, and databases. Some of the tool developed have shown to be a key player in this scenario, both in the identification of check worthy claims and in the verification of their veracity. If on one hand these tools

provide a valuable support, on the other (i) they strongly rely on the traditional channels of communication (e.g. TV, radio, etc.) [47, 7] and (ii) they appear as a black box which does not provide transparency, a core aspect of the fact checking process. The need to find a cost effective way to engage fact checkers in the developing of a social media monitoring system, providing them more control on tool usage and understanding of its features output, is been the main goal of our research.

Scope. This project is in collaboration with RMIT ABC Fact Check[1], an Australian Fact Checking organization. The team is mainly composed by journalists, editors, and designers and their goal is to test claims made in the public domain by politicians, public figures and advocacy groups. They constantly monitor broadcast, print and social media, as well as Parliament, for check-able claims made in the public domain. Fact-checkers choose what claims to check looking at (1)the content of the sentence and then decide if it is eligible to be considered as a claim, and (2)who is saying it, who is repeating it, and so on. The two things together are an indicator of how interesting, important, and influential the claim is. This is the first step in the fact checking process, and also the one that has attracted many researches. There have been, in fact, many attempts to automatize it and we actually have now powerful tools (e.g. *Claimbuster* [23] , or CheckThat!Lab [3]) that usually provide a ranked list of check-worthy claims retrieved from different platforms (i.e. tv, radio, social networks) and mainly relying on machine learning and natural language processing technologies. Our goal is not to provide an automatic tool, but a systematic access to the huge amount of data potentially available from social networks, and guide the experts in the identification of claims in a transparent way.

We chose Twitter as social media to monitor and potentially leading to claims/topic relevant for the fact checkers. Twitter, though not as wide-reaching as Facebook, is an important element of a campaign's digital strategy. Candidates, parties, journalists, and a steadily increasing share of the public are using Twitter to comment on, interact around, and research public reactions to politics. [25] On Twitter, most user accounts are publicly visible and accessible even for non-registered audiences. Its usage is centered around topics and the

[1] https://www.abc.net.au/news/factcheck/

retweet feature facilitates the diffusion of political information beyond the direct follower network [48]. In contrast, most accounts on Facebook are private and its usage is based on one-way or reciprocal friendship ties. Information travels less fluidly through this medium, also due to the extensive algorithmic filtering of contents. [44]. Among the most active Twitter users are prime targets of campaigns like political elites and influentials. Journalists, for instance, regard Twitter of higher value for news reporting while using Facebook primarily for private purposes [35]. These findings make Twitter a suitable platform to be monitored and analysed to accomplish fact checkers goal.

Research Questions. Social media for news consumption is a double-edged sword. On the one hand, its low cost, easy access, and rapid dissemination of information lead people to seek out and consume news from social media. On the other hand, it enables the wide spread of fake news, i.e., low quality news with intentionally false information. Fact checking organizations are more and more needed in this scenario, and this research aims to understand how to better integrate data analytics tools in their daily work. After analysing the different steps of the fact checking process - claim identification, verification and verdict- we chose the first step as our target. Then we proceed with the monitoring of Twitter as our target social network.

Therefore we have worked to address the following questions:

(1) What is the most cost effective way to integrate data analysis in their daily work, engaging them in the development and giving them a transparent control of the outcome of the analysis?
(2) What analysis can be done to guide them towards claim identification on social media, and not to replace them with automatic tools?

Our proposed solution is a two-step process. We first designed the infrastructure for crawling the data from Twitter, pre-process it and finally indexing it for allowing fast full text search on tweets fields. Then we started the collaboration with the fact checkers based on a *lean methodology* following the principle of 'Continuous Improvement' and avoiding waste of time and resources.

Summary of Contributions. In this thesis, we present an ongoing work to develop *Watch n' Check*, a tool that can be used by fact-checking experts to facilitate their access to relevant information in volumes of data generated by Twitter. Such tool can be beneficial as, (i) it allows experts access to large amount of information streams published in real-time; (ii) it can enable an efficient identification of trending topics and monitoring the changes in the news propagation dynamics; and (iii) it improves the impartiality in the process of targeting the statements/news to be checked.

Our Watch n' Check prototype is being developed in collaboration with fact-checking experts from RMIT ABC Fact Check. The aim of our collaboration is to identify a cost effective methodology to interact with the experts and identify key functionalities which would inform the design and development of an information access tool to assist fact checkers with identifying claims in social media.

1.1 Structure of the Thesis

The thesis is structured as follow:

Chapter 1. The introduction leads the reader from the general subject area to the particular topic of inquiry. It establishes the scope, context, and significance of the research being conducted by summarizing current understanding and background information about the topic, stating the purpose of the work in the form of the research problem supported by a set of questions, explaining briefly the methodological approach used to examine the research problem, highlighting our contribution and the potential outcomes of the research, and outlining the remaining structure and organization of the paper.

Chapter 2. Literature review about books, scholarly articles, and any other sources relevant to our area of research, providing a description, summary, and critical evaluation of these works in relation to the research problem being investigated. It first analyzes related works and research about social media and spread of fake news, then provides further details about fact checking organizations and automated fact checking tools, thus social media monitoring

tools. It also gives more detail about the technology underlying our tool, explaining some of the key features they offer, necessary to accomplish our goals.

Chapter 3. The methods section describes actions we have taken to investigate our research problem, answering two main questions: How was the data generated and collected? And, how do we develop and integrate the analysis on the data, always engaging fact checkers experts in the process?

Chapter 4. The results section reports the findings of our study based upon the methodology applied to gather information. The results section states the findings of the research arranged in a logical sequence.

Chapter 5. The purpose of the discussion is to interpret and describe the significance of the results in light of what was already known about the research problem being investigated and to explain any new understanding or insights that emerged as a result of the study of the problem. Plus, it further describes the limitation of our work, suggesting new potential developments.

Chapter 6. The conclusion is a synthesis of the key points of the thesis, intended to help the reader understand how this research answered our initial research questions.

CHAPTER 2

Background

This chapter provides two sections. First a literature review on related work about Automated Fact Checking systems and their connection with Fact Checking organizations. We will further describe social media and how they are currently monitored and what analysis are performed on them, further describing the technologies underlying. The second section describes in detail the technologies that we have used for our research, considering our needs and goals.

2.1 Related Work

In this section we describe the state of the art of the automated fact checking systems currently in use from some of the Fact Checking companies. Therefore, we analyse some of the Social Media monitoring tools. Both subsections constitutes the basis for our research development.

2.1.1 Social Media as News Source

Social media has become a part of the fabric of our daily lives. Facebook, Twitter, YouTube and many other social networking sites allow users to share and interact with online content and to connect with like-minded people. Rapid dissemination, amplification of content, and the ability to lead informal conversations are the strengths that make it a powerful tool in many professional contexts [34]. One of the field where social media have a vital role is the news seeking and consumption. As of August 2018, two-thirds (68%) of Americans report that they get at least some of their news on social media, registering an increase from the previous years –according to a survey from Pew Research Center [41]. Europe presents a

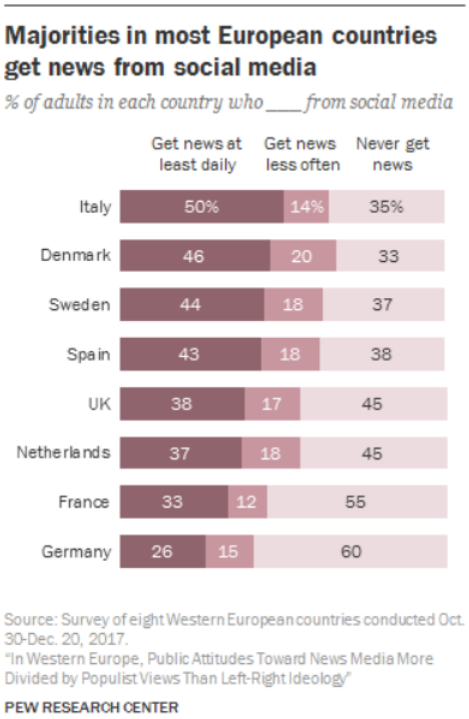
similar situation, even though every country has a slightly differentiated way to consume news. In fact, accordingly to the survey from Pew Research Center [41], in six of the eight countries surveyed, more than half say they ever get news from social media. And much of this news use occurs on a daily basis, especially in Italy, where half do so at least once a day [29]. Australian news consumers in general access news less often and have lower interest in it compared to citizens in many other countries. The survey finds almost half (48%) of Australian news consumers access news once a day or less, whereas the global average across the 38-countries is one-third (34%) [16]. New research from Roy Morgan reveals over 13 million Australians (65.6%) now say TV is a main source of news, followed by the Internet used by 11.7 million Australians (57.8%) and the leading source of online news is social media used by 7.5 million (36.7%) [45].

2.1.2 Dark Side of Social Networks as News Source: Fake News

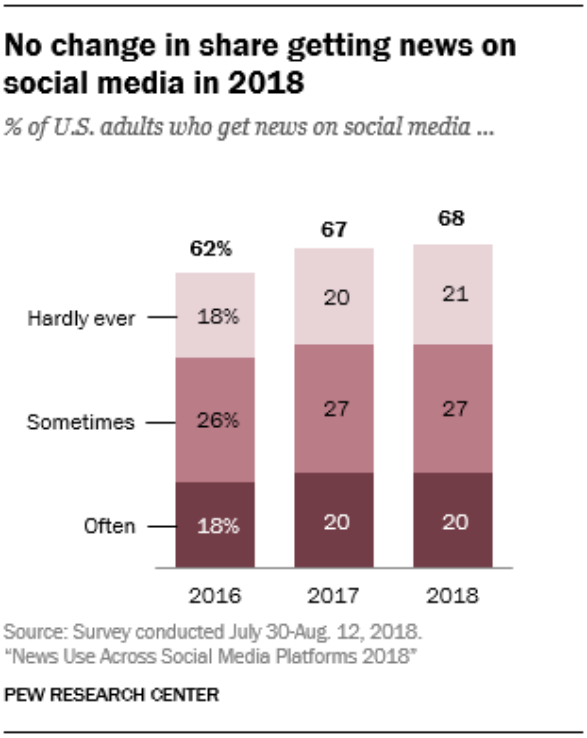
Social media has made access to and exchange of news and information in written, verbal and visual form, very convenient and easy. On the other side, the authenticity of information has become a longstanding issue and the spread of fake news has been a growing phenomenon.

Definition. There is no agreed definition of the term "fake news", but the most widely adopted in recent studies [10, 31, 37] is the following: *Fake news is a news article that is intentionally and verifiably false and could mislead readers.* [2].

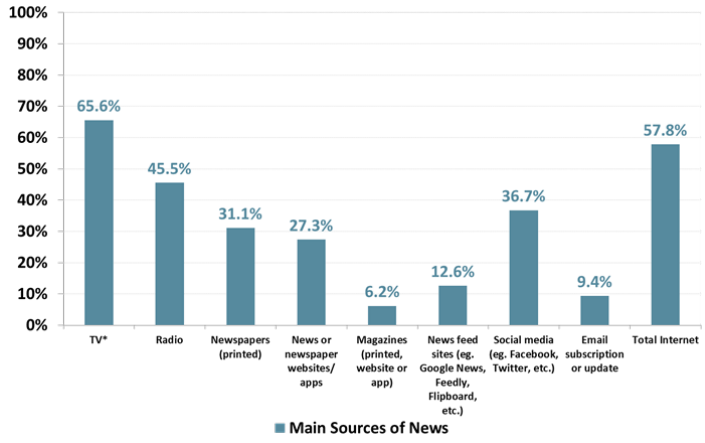
Common patterns. Fake news itself is not a new problem since the sensationalism of not-so-accurate eye catching headlines aimed at retaining the attention of audiences to sell information has persisted all throughout the history of all kinds of broadcast information. The rise of web-generated news on social media makes fake news a more powerful force that. On one side they reflect the basic social and psychological theories related to fake news, on the other side they introduce more advanced patterns strictly related to social media [42]. In fact, traditional fake news mainly targets consumers by exploiting their individual vulnerabilities based on two major factors (i) Naıve Realism: consumers tend to believe that their perceptions of reality are the only accurate views, while others who disagree are regarded as uninformed,



(A) News consumption in social media in Europe



(B) News consumption in social media in US



(C) News consumption in social media in Australia

FIGURE 2.1. News consumption across the World from The Conversation [16]

irrational, or biased [39]; and (ii) Confirmation Bias: consumers prefer to receive information that confirms their existing views [33].

Fake news on social media. Fake News spread on Social Media is enhanced by two particular factors, not existing in other communication channels: (i) Propaganda made by malicious account and (ii) Echo Chamber Effect [42].

Social media users may be malicious and not even real humans, such as social bots, cyborg users, and trolls. A social bot refers to a social media account that is controlled by a computer algorithm to automatically produce content and interact with humans (or other bot users) on social media [14]. Trolls are real human users who aim to disrupt online communities and provoke consumers into an emotional response and, finally, cyborg accounts are registered by human as a camouflage and set automated programs to perform activities in social media. All of them have a common goal, which is to thrive misinformation and mislead people [9]. For example, studies shows that social bots distorted the 2016 U.S. presidential election online discussions on a large scale, and that around 19 million bot accounts tweeted in support of either Trump or Clinton in the week leading up to election day [4].

Research shows that the echo chamber in communities become the primary driver of information diffusion that further strengthens polarization in news consumption [11]. Social media provides a new paradigm of information creation and consumption. Users are selectively exposed to certain kinds of news because of the way news feed appear on their homepage in social media. For example, users on Facebook always follow like-minded people and thus receive news that promote their favored existing narratives, creating the Echo Chamber effect mentioned above [38]. The following psychological factors are the reason why people are more likely to not recognise fake news [36]:

(1) *social credibility*. People tend to believe that a source is credible if other people perceive the source as credible, especially if they have not enough information to assess their falseness
(2) *frequency heuristic*. The more frequent and spread is a news, the more likely it will be considered as true, even if it is not. Studies have shown that increased exposure to an idea is enough to generate a positive opinion about it [50, 51].

Impact on people. The reach and effects of information spread on social networks occur at such a fast pace and so amplified that distorted, inaccurate or false information acquires a tremendous potential to cause real world impacts, within minutes, for millions of users. A clear example is the new avalanche of misinformation that has accompanied the COVID-19 pandemic. Unreliable and false information is spreading around the world to such an extent that some commentators are now referring to as a 'infodemic' [1], or 'disinfodemic'. The impact of fake news is so dangerous that this phenomenon is putting lives at risk. Iran has counted hundreds of deaths over the false belief that drinking methanol cures coronavirus, 5,011 people had been poisoned and about 90 people had lost their eyesight or were suffering eye damage from the alcohol poisoning [32]. Fake news have also contributed to the escalation in racially motivated abuse towards people from Asian backgrounds, considered as the spreaders of the virus. Two Asian students were assaulted in Melbourne as they went to the supermarket [40]. The conspiracy against 5G as main contributor of the spread of the virus have been amplified on social media platforms. In Britain dozens of phone masts and other pieces of critical communications infrastructure have been vandalised since the beginning of April. Footage of UK telecommunication engineers being harassed by members of the public has surfaced online [8].

2.1.3 Fact Checking

In this panorama, the need of truth-seeking organizations and methods has gained prominence as a fundamental process for revitalizing news broadcasting and consumption.

Definition and common patterns. Fact checking is the task of assessing the truthfulness of claims made by public figures such as politicians, news media, pundits, commentators [47]. An increasing number of fact-checking outlets exist, and across different countries, different organisational forms, and different self-identified orientations, but globally follow the same missions: assessing the truth of political claims and giving citizens information to make political choices. Fact-checking may be done in-house by the publisher or it can be analyzed by a third party via external fact-checking [20]. According to a report surveying the landscape

of fact-checking outlets in Europe [7], the majority of the fact checking organizations are composed by journalists, activists, policy experts, and with just a very small percentage of technologists.

Methodology and ethics of fact-checking. Fact-checking is composed of three phases:

(1) Finding fact-checkable claims by scouring through legislative records, media outlets and social media. This process includes determining which major public claims (a) can be fact-checked and (b) ought to be fact-checked.
(2) Finding the facts by looking for the best available evidence regarding the claim at hand. This phase is a detailed analysis of the data available and related to the claim, ans and strongly depends on the context. For example, the statement "We've always been very, very careful to prioritise Australians and Australian jobs" made by the Minister for Population, Cities and Urban Infrastructure Alan Tudge is been identified as 'Half True' by RMIT ABC Fact Check [1].The context is crucial to the final verdict, the fact checkers need to consider that it is made by a Australian politician, so they have to consider the data about that country. Furthermore, time is also important since the various comparisons usually refer to time-frames related to the time a claim is made [47].
(3) Correcting the record by evaluating the claim in light of the evidence. It would natural be to consider it as a binary classification: True or False. It is often the case that the statements are not completely true or false, so it becomes an ordinal classification task [17]. Accordingly to the European survey, some of the organization use a scale indicating degrees of truth, others provide a label representing different kind of factual errors, few others don't have rating systems, but only writing conclusion.

Organizations across the globe. External post hoc fact-checking organizations first arose in the US in the early 2000s, though the concept first grew in relevance and spread to multiple countries during the 2010s. Although they follow a common goal, they all slightly differ in their fact-checking methodology, as we can notice from some of the most famous websites:

[1] https://www.abc.net.au/news/2020-06-03/fact-check-migrant-workers-457-temporary-visas/12299180

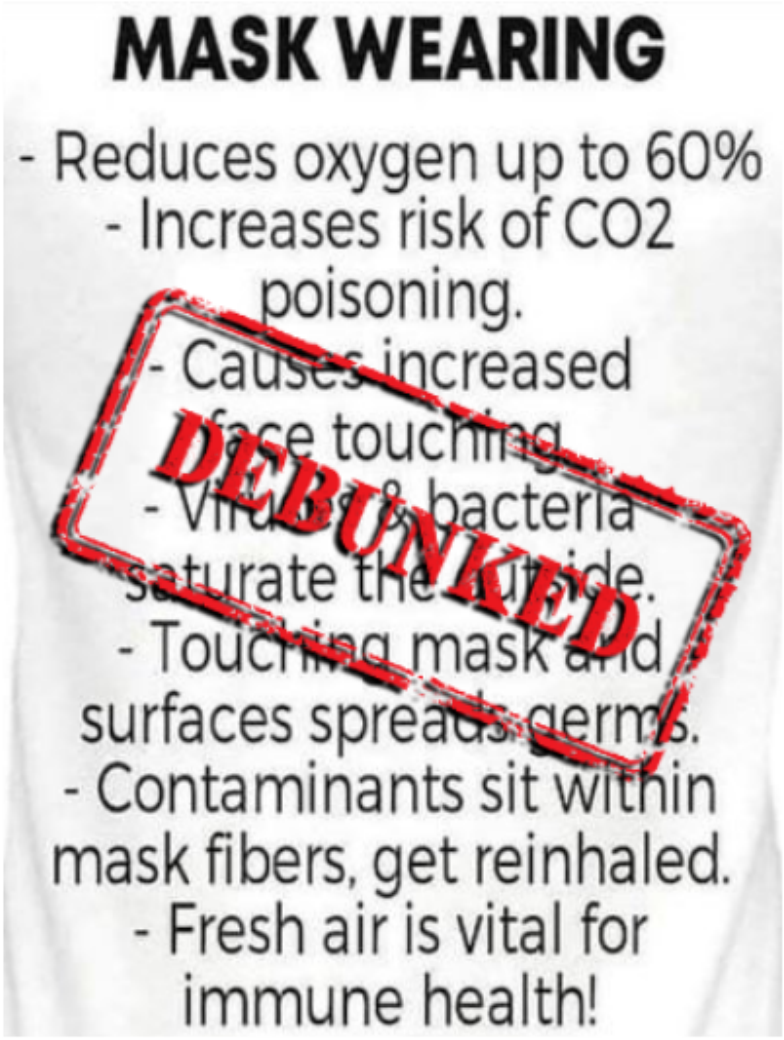
- RMIT ABC Fact Check (accessible at `https://www.abc.net.au/news/factcheck/about/`)
- PolitiFact's "The Principles of PolitiFact" (accessible at `https://www.politifact.com/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-`
- Pagella Politica's "Metodologia" and "Come funzioniamo" (accessible in Italian at `https://pagellapolitica.it/progetto/index`)
- Chequeado's "Método" (accessible in Spanish at: `http://chequeado.com/metodo/`)

The International Fact-Checking Network (IFCN)[2] has developed a code of principles that guide fact-checking organisations in the application of standards and principles in their daily work.

Funding. A worldwide survey by the Poynter Institute in May 2016 found similar results: nearly 45% of fact-checking budgets were below $20,000, while about 29% were greater than $100,000. Full-time fact-checking by professional researchers in a wealthy country is an expensive proposition. Full Fact, supported mainly by foundation grants and individual donations, publicly estimates that costs will reach £600,000 in 2016. The Conversation UK, also a registered charity, reports 2016 expenditures approaching £1,000,000. It is evident that the fact checking outlets have to face significant financial challenges in order to support their goals, mainly relying on media industries and charitable foundations.

RMIT ABC Fact Check. RMIT ABC Fact Check is a partnership between RMIT University and the ABC combining academic expertise and the Australian journalism, working together to inform the public with an independent non-partisan voice. It is funded jointly by RMIT University and the ABC. The ABC is a publicly funded, independent media organisation, and therefore RMIT ABC Fact Check is accountable to the Australian Parliament. Fact Check goal is to test claims made in the public domain by politicians, public figures and advocacy groups that can be tested against available data at the time they are made. All verdicts fall into three colour-based categories: in the red, in the green or in between – red being a negative ruling, and green being a positive.

[2] https://ifcncodeofprinciples.poynter.org/



Claims on mask wearing were debunked by fact checkers. *(Supplied)*

FIGURE 2.2. Example of debunked fact from RMIT ABC Fact Check [6]

The team is mainly composed by journalists, editors, and designers. They constantly monitor broadcast, print and social media, as well as Parliament, for check-able claims made in the public domain. Once the director approves a claim, one of the researchers contacts experts in the field to seek their opinion and guidance on available data. The expert opinion and data is written into a draft, which is then reviewed by the chief fact checker, who identifies problems, and challenges the researcher on anything that they might have missed. The chief fact checker also scrutinises all sources and makes sure the draft is consistent with what the

data says. The researcher continually reworks the draft based on this feedback, and once the chief fact checker is satisfied, the team discusses the final verdict. These discussions are rigorous and much thought is given to the verdict word and the colour that will be used, which is an important part of how they inject nuance into verdicts. The online editor then prepares the final product, which is once again checked by the chief fact checker for any inaccuracies which may have crept up during the editing process. Once the director signs off on the finished draft, it's ready to be released to the world.

2.1.4 Automated Fact Checking

The recently increased focus on misinformation has stimulated research in fact checking, the task of assessing the truthfulness of a claim. Research in automating this task has been conducted in a variety of disciplines including natural language processing, machine learning, knowledge representation, and databases. Automated Fact Checking systems provide so far a solution that address one or more of the three core elements of Fact Checking [19]: (1) Identification of false or questionable claims, monitoring media and political source, identifying factual statements and prioritising claims to check. (2) Verification of the truthfulness/falseness of the claim checking it against authoritative sources or existing fact-checks. (3) Corrections of the claim across different media to audiences exposed to misinformation, flagging the falsehoods and providing contextual data.

Over the last several years, AFC has seen an exponential growth of research literature in its field, especially in the intersection between Artificial Intelligence and practical experiments provided by Fact Checkers. An example is the Fake News Challenge, which aims to explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem and significantly automating parts of the procedure human fact checkers use today to determine if a story is real or a hoax [3].

[3] http://www.fakenewschallenge.org/

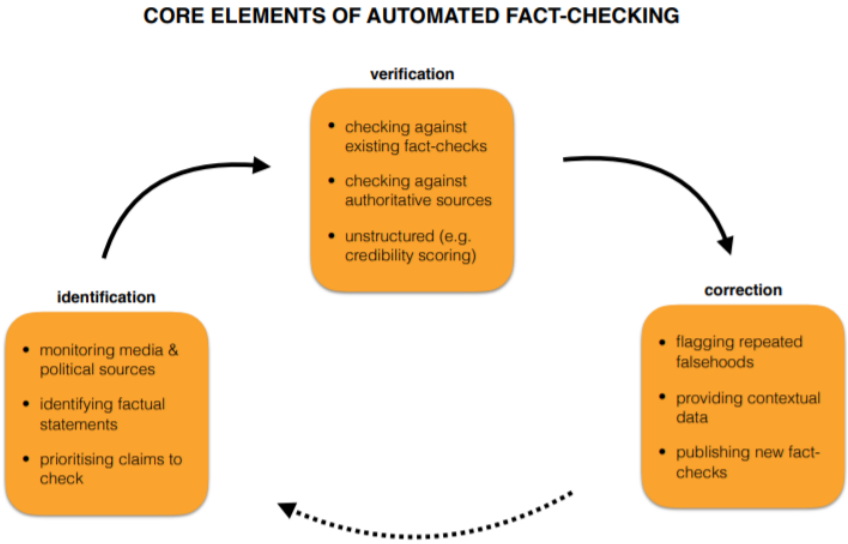


FIGURE 2.3. Core Elements of Automated Fact-Checking

Identification of claims. The first step for an Automated Fact Checking tool is the monitoring of the various forms of public discourse, like speeches, news, reports, social media. This is a task that involves scraping data from media, political pages, online and traditional channels. Furthermore, monitoring all these sources consistently and stably is a difficult engineering challenge in term of scaling and fault tolerance.

An example of tool for monitoring TV and Radio is the Microsoft Audio Video Indexing Service (MAVIS) that uses state of the art speech recognition technology developed at Microsoft Research to enable searching of audio and video files with speech. MAVIS automatically generates closed captions and keywords which increase accessibility and discoverability of audio and video files with speech content [4].

Some of the social media like Twitter offer API to get access to their content, and it will be further discussed in the second section of this chapter. Other social media like Facebook don't provide any API, and only Facebook itself can access user's posts.

The second challenge is how to spot claims in the data collected. The most common approach relies on a combination of natural language processing and machine learning to identify and

[4]https://www.microsoft.com/en-us/research/project/mavis/

prioritise claims to be checked. When human factcheckers choose what claims to check, they look at two things. First, they look at the content of the sentence and then decide if it is eligible to be considered as a claim, Secondly, an important factor is represented by who is saying it, who is repeating it, and so on. The two things together are an indicator of how interesting, important, and influential the claim is. The content approach is harder for computers, and it usually relies on machine learning and natural language processing techniques.

ClaimBuster represents a good example of tool able (i) to spot claims in political discourses and (ii) to suggest an order of priority for tackling them [21]. It scores the 'checkability' of statements. Sentences which share most features with sentences previously marked as check-worthy get higher probabilities of being check-worthy. It is able to do so because it was 'taught' by human researchers, on the basis of the past 30 presidential debates, the difference between a non-factual sentence, an unimportant factual sentence (such as, "Tomorrow is election day.") and a check-worthy factual sentence. Compared to human analysts, it correctly isolated 74% of check-worthy statements. It is a tool that cannot alone do the work of human fact-checkers, but one that can greatly assist them with a small part of it [28].

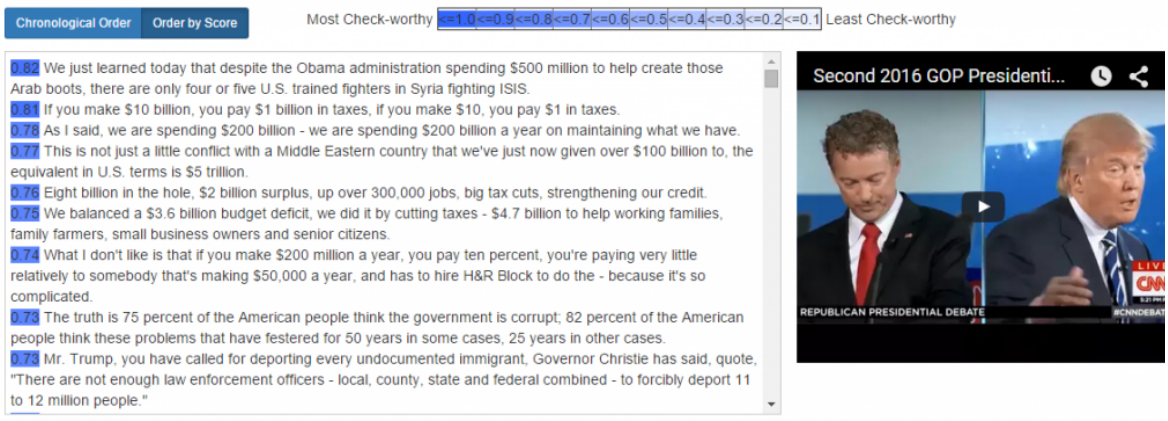


FIGURE 2.4. Claimbuster Output Example

Another cutting-edge solution in identification of claims is represented by `Full Fact`. The system is split into two main tools, Live and Trends [5]. `Trends` tracks every repetition of a claim certified as wrong, as well as where it comes from, so it can keep track of who or what is spreading misinformation into the world. `Live` spots claims in TV subtitles that have

[5] https://fullfact.org/automated

been fact checked before and automatically pulls up the most recent fact checked articles in response. It also spots claims that haven't been fact checked before - but reliable data exists for - and creates fact checks on the spot using that data.

Combining the content and context approaches —just as human factcheckers do— is the main challenge for automated fact checking. In fact, the same claim can be made in many different ways. A fully automated factchecking system should be able to identify different phrases which make the same claim, and distinguish very similar phrases that make different claims, as humans can. Also, it should be able to recognise a claim made over more than one sentence. Paraphrasing in factchecking representes the main obstacle since precise wording can matter so much conclusion.

Verifying Claims. Two primary approaches to automatic verification are (i)matching statements to previous fact-checks and (ii) consulting authoritative sources.

Matching statements against a library of claims already checked by one or more fact-checking organizations accelerates the process and also accelerate the production of new fact-checks. Some famous libraries are provided by FactCheck.org ([6]), Politifacts [7] and other fact checking companies that use Share the Facts. Share the Facts implements the ClaimReview schema, an open standard for coding the different components of a fact check, such as the claim and the verdict, in a machine-readable way. It allows fact checking organisations to tag their articles in such a way that allows third parties such as Google, Facebook, Bing and Youtube to analyse and sort the fact checking data accordingly in their indexes [8]. This is not effective in the case the claim is not exactly matching with the one in the library, because even subtle changes in the wording, timing, or context of a claim can make it more or less reasonable, and potentially leading to false positives.

A greater challenge at the centre of current research is to verify claims against the same kinds of original information sources relied on by human fact checkers. It requires the AFC system can recognise the kind of data called for, and that the data are available from an authoritative

[6] https://www.factcheck.org/
[7] https://www.politifact.com/
[8] https://wordpress.org/plugins/claim-review-schema/

source in a form a machine can collect and use. This could potentially expand the range of statements which can be checked automatically. In practice, fully automatic verification today remains limited to experiments focused on a very narrow set of cases of mostly statistical claims that can be verified against specific public statistics, published in a structured way and easily accessible through API.

Another avenue of research involves less structured or "non-reference" approaches to verification. In fact, rather than looking up a specific authoritative reference, these methods rely on a variety of content or network-related characteristics to make inferences about the likely truthfulness of a claim. These range from stylistic features like the kind of language used in a social media post (Style Based detection), or the way a particular claim or link propagates across the internet (Propagation based detection) [49]. This shifts the problem from determining veracity to scoring reliability, and consequently cannot be a substitute for assessing the factual accuracy of individual statements.

Correction and publishing. This stage of factchecking is still far from being automatized. Communicating the result of a fact checking is the last, and also a crucial step of the entire process. Fact checkers need to engage the public and incentive it to read their contents. One good graphic might be worth a thousand words. All their research and analysis must be simplified and it has to be authoritative and recognisably fair for people on different sides.

Finally, automated systems for claim identification and verification can be very useful as supportive technology for investigative journalism, as they could provide help and guidance, thus saving time and resources [18, 22, 23, 46].

2.1.5 Social media monitoring tools

Social networks are important means for communication, engaging millions of users around the world. In the recent years it is been particularly interesting for all kind of enterprises being present and aware of what is discussed on those new communication channels. In fact, they usually provide meaningful insight on real customers opinions, complaints, questions

and desires in a real-time and scalable way. They appear as a wealth source of information available online for free in the form of user-generated content, that in very few years overcame the traditional methods to listen to customers and to communicate [30].

As a result, social media monitoring tools (SMM) and platforms have emerged to address the need of enterprises for analysing their customers activity online. They can be relevant for a broad variety of stakeholders: businesses use SMM for online reputation of the own brands, products and services, marketing can use the insight to control the impact of campaigns, product and innovation management could use their customer's opinions to create new ideas and offer a product or service that meets better the needs of their users.

In a market study on social media monitoring tools of Fraunhofer IAO [26] following application fields have been specified:

- reputation-management;
- event detection, issue- and crisis-management;
- competitor analysis;
- trend- and market-research plus campaign-monitoring;
- influencer detection and customer relationship management;
- product- and innovation-management.

As we can notice, these tools aim to provide insights for other scenarios than fact-checking. Our contribution is to explore the connection between the two fields, but first we need to identify what are the concepts behind a social media monitoring tool.

Main features. Social media monitoring tools have common patterns and features that they provide for their users. Fan and Gordon (2014) [13] propose a process consisting of three steps: "capture", "understand", and "present". The authors state that the step of capture consists of gathering the data and preprocessing it, whereas pertinent information is extracted from the data in this step. Afterwards, noisy information, if existing in the data, should be removed. However, the core of this step consists of applying a key technique, such as a sentiment analysis or social network analysis, for understanding the data. In the last step the

findings should be summarised and presented [13]. Stieglitz et al. (2014) [43] also propose a framework for social media analytics (SMA), which is the most accepted one in information systems, based on the citations of the paper in IS literature. The authors describe the SMA process as consisting of three steps (see Fig. 2.5).

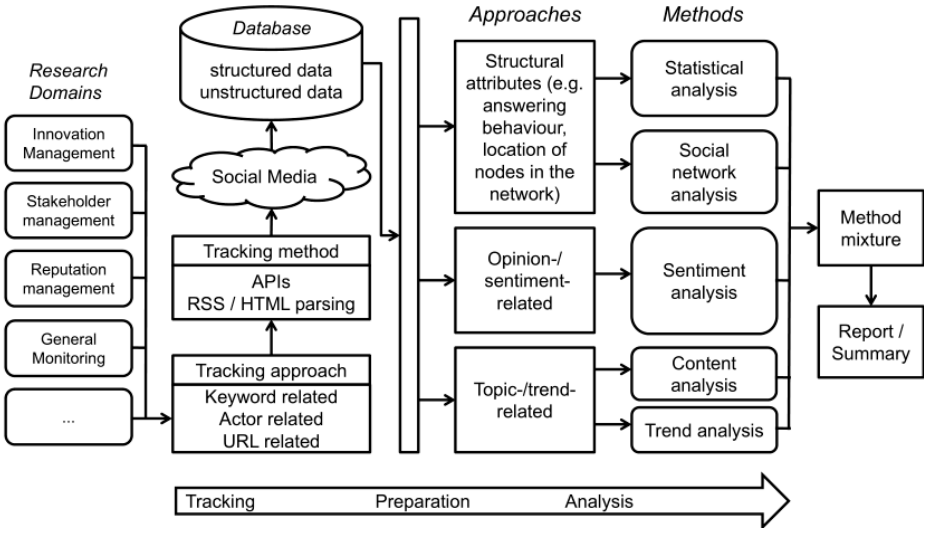


FIGURE 2.5. The Social Media Analytics Framework from Social MediaAnalytics: An Interdisciplinary Approach and Its Implications for Information Systems [43]

Underlying technologies. A social media monitoring tool has to provide particular technological features in order to be effective and accomplish the goals and features listed above. In the research 'An approach for evaluation of social media monitoring tools' they provide a list of characteristics that the tool is required to have:

(1) it needs to specify a listening grid which includes (i) what are the channels that are monitored (e.g. blogs and micro-blogs, social networks, video and image websites, (ii) which countries and languages the tools provide support for and (iii) the topics relevant to the enterprise
(2) it has to be near real-time
(3) it provides access to historical data in order to compare the current metrics and reports related to the monitored topic with any previous state of it.
(4) it offers API for the integration with other internal/external tools

(5) it enables topic detection and sentiment analysis for finding valuable information in user-generated data, using text analytics, and machine learning elements, such as latent semantic analysis, support vector machines, natural language processing.

User Interface. The users need to have an interface that shows the retrieved information and insights in a meaningful and concise way. Dashboards are often used for this purpose, given their natural predisposition to integrate information from multiple components into a unified display in offering graphical representation of the raw data in the form of charts, listings, and historical graphing of queries and phrases. Some tools allow advanced configuration options for filtering language, region, media type, or organize the results found and enable users to download the results of their tool's analysis in different formats such as excel workbook or PDF.

2.1.6 From traditional media to social networks: `CheckThat!Lab`

CheckThat! Lab [9] represents the bridging between AFC and SMM tool and it is the very first tool that provides features that span the full verification pipeline on Twitter claims. It is currently in its third edition of the lab, officialized in January 2020. The 2018 edition of CheckThat! focused on the identification and verification of claims in political debates [3] whereas the 2019 edition also focused on political debates in conjunction with a closed set of Web documents to retrieve evidence from [12]. The third edition focuses on Twitter data, and it is organised in different tasks as shown in Fig. 2.6. First, they collect data from Twitter and manually organise it in topics. The topics are then the input of the `CheckThat!` system and it returns a list of tweets for each topic, ranked on their check-worthiness.

The second phase aims to check the previous selected claims against a data-set of already verified claims provided from Politifact[10] and Snopes[11]. The system create a list of verified

[9] CheckThat! at CLEF 2020: Enabling the Automatic Identification and Verification of Claims in Social Media

[10] https://www.politifact.com/

[11] https://www.snopes.com/

claims ranked on their relevance with the initial claim to verify. The relevance takes in consideration text similarities and entailment.

If the claim has not been verified, it proceeds with the third phase which provides a rank of the top-m websites that match the topic and the context of the claim to verify and offers a source of evidence.

Task 4 is a binary classification problem which, given a check-worthy claim from a tweet and a set of potentially relevant Web pages, it predicts the veracity of the claim.

`CheckThat!` is the first shared task that addresses all steps of the fact-checking process on Social Media. The third editions, as well as all the editions of the lab, are based on one of the AFC pillar: an automated system could automatically identify check-worthy claims, make sure they have not been fact-checked already by another fact-checking organization, and then present them to a journalist for further analysis in a ranked list. Additionally, the system could identify documents that are potentially useful for humans to perform manual fact-checking of a claim, and it could also estimate a veracity score supported by evidence to increase the journalist's understanding and the trust in the system's decision.

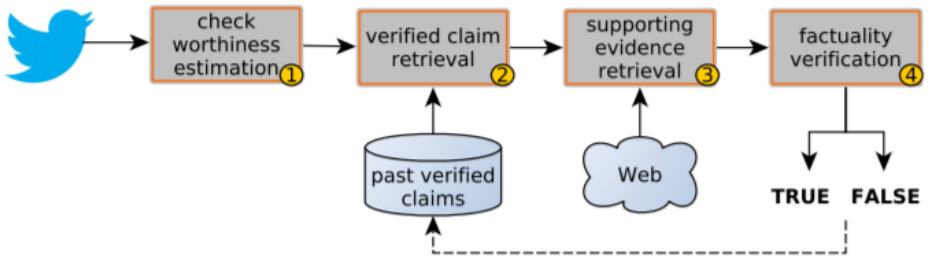


FIGURE 2.6. CheckThat! Information verification pipeline

2.2 Technologies

This section provides details about the technologies used to interact with the fact checkers and develop *Watch n' Check*, addressing some of the main features they offer and how they address our needs.

2.2.1 Twitter Crawler

In this chapter we briefly describe the technology that allows the extraction of tweets from Twitter. This is the first step of the entire analysis. The goal is to provide an understanding of the entire process, from the collection of the data, how the connection with Twitter is managed and what kind of data are retrieved.

Tweets structure. A tweet is a string of at most 140 characters, often linked to an image or URL. Related to the single tweet, some other information are available, like the number of people who have retweeted it and the number of likes to the tweet itself. Both data give an idea on how popular is that tweet.

Beyond what the user can see on the platform and its front-end, Twitter collect many other information on every tweet, which we will refer to as *metadata* and that can provide valuable insight for more advanced analysis. [12]. Attributes such as who posted, at what time, whether it's an original Tweet or a Retweet, and an array of first-class objects such as hashtags, mentions, and shared links. For the account that posted, there is a User (or Actor) object with a variety of attributes that provide the user's Profile and other account metadata. Profiles include a short biographical description, a home location (free form text), preferred language, and an optional web site link.

Therefore we can differentiate *data* and *metadata* as the following:

(1) Data: Information of the post visible to the user through the graphical interface of the social network, they are also the information on which a user can actually modify

[12] https://developer.twitter.com/en/docs/tweets/data-dictionary/guides/tweet-timeline

FIGURE 2.7. Tweet Example

(2) Metadata: Information not visible to the user, but collected by requests to the API.

API: Application Programming Interfaces. Twitter allows developer and user to a programmatic access to its metadata, through the use of API. As per Wikipedia's Definition [13] of API: In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building software and applications. API is a kind of interface which has a set of functions that allow programmers to access specific features or data of an application, operating system or other services without having to know how they're implemented..

[13] https://en.wikipedia.org/wiki/Application_programming_interface

Web API. Web API as the name suggests, is an API over the web which can be accessed using HTTP protocol. While "web API" historically has been virtually synonymous with web service, the recent trend has been moving away from Simple Object Access Protocol (SOAP) based web services and service-oriented architecture (SOA) towards more direct representational state transfer (REST) style web resources and resource-oriented architecture. Twitter, in fact, provides REST APIs that allows a programmatic access to read and write data using which we can integrate twitter's capabilities into our own application.

REST. The term Representational State Transfer (REST) refers to an architecture that aims to creating network applications, based on a stateless client-server communication protocol. It is important to specify that this architecture is independent from the protocol because it interfaces with it, without identifying it [15]. Therefore, the fundamental idea of such approach consists in using a communication protocol (e.g. HTTP) to make two machines communicate on a network. This approach is hence identified as an alternative, to mechanisms such as Remote Procedure Calls (RPC) and Web services (e.g., WSDL, SOAP4). As a matter of fact, applications based on this approach usually use the HTTP protocol for all Create, Read, Update, Delete (CRUD5) operations.

Endpoints. One of the basic principles of the REST architecture is that each resource must be identified by a unique URI, which corresponds to the endpoint. Endpoints are important aspects of interacting with server-side web APIs, as they specify where resources lie that can be accessed by third party software. Usually the access is via a URI to which HTTP requests are posted, and from which the response is thus expected.

Twitter APIs. Twitter offers a range of REST API, each one with its unique endpoint and function, collecting a variety of data that can be suitable for different purposes.

In our research we chose to use *Sampled stream v1* that allows developers to stream about 1% of all new public Tweets as they happen. There are different API, like *Tweet metrics v1* that allows developers to access engagement metrics for any Tweet or list of Tweets from owned/authorized accounts. By metrics, we mean the total count of impressions, Retweets,

Quote Tweets, likes, replies, video views, and video view quartiles for each Tweet specified in the request [14].

Authentication. Twitter APIs handle enormous amounts of data. The way they ensure this data is secured for developers and users alike is through authentication [15]. There are a few methods for authentication, but the most common methods used by the Twitter Developer Platform are OAuth 1.0a and OAuth 2.0 Bearer Token. OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.

OAuth 1.0a: Application-user authentication. OAuth 1.0a [16] allows an authorized Twitter developer app to access private account information or perform a Twitter action on behalf of a Twitter account. You will need user-authentication, user-context, with an access token to perform the following:

- Post Tweets or other resources
- Search for users
- Use any geo endpoint
- Access Direct Messages or account credentials
- Retrieve user's email addresses

OAuth 2.0: Application-only authentication. Application-only authentication doesn't include any user-context and is a form of authentication where an application makes API requests on its own behalf. This method is for developers that just need read-only access to public information. This means that we can perform actions such as:

- Pull user timelines
- Access friends and followers of any account
- Access lists resources

[14] https://developer.twitter.com/en/docs/labs/tweet-metrics/overview
[15] https://developer.twitter.com/en/docs/basics/authentication/overview
[16] https://developer.twitter.com/en/docs/basics/authentication/oauth-1-0a

- Search Tweets

OAuth 1.0a and OAuth 2.0: Comparison. La principale differenze tra le due è che OAuth 1.0 requires cryptographic signing of each request. With OAuth 2.0 Bearer tokens, it is possible to quickly make API calls from a cURL command. The access token is used instead of a username and password. For example, before OAuth, you may have seen examples in API docs such as:

```
curl --user bob:pa55 https://api.example.com/profile
```

With OAuth 1 APIs, it become no longer possible to hard-code an example like this, since the request must be signed with the application's secret. Some services such as Twitter started providing "signature generator" tools in their developer websites so that you could generate a curl command from the website without using a library. For example, the tool on Twitter generates a curl command such as:

```
curl --get 'https://api.twitter.com/1.1/statuses/show.json
--data 'id=210462857140252672'
--header'Authorization:OAuth
   oauth_consumer_key="xRhHSKcKLl9VF7fbyP2eEw",
oauth_nonce="33ec5af28add281c63db55d1839d90f1",
oauth_signature= "oBO19fJO8imCAMvRxmQJsA6idX3D",
oauth_signature_method="HMAC-SHA1",
oauth_timestamp="1471026075",
oauth_token="12341234-ZgJYZOh5Z3ldYXH2sm5voEs0pPXOPv8vC0mFjMFtG",
oauth_version="1.0"'
```

With OAuth 2.0 Bearer Tokens, only the token itself is needed in the request, so the examples again become very simple:

```
curl -X GET -H "Authorization: Bearer \$BEARER\_TOKEN"
   "https://api.twitter.com/labs/1/tweets/stream/sample"
```

2.2.2 ElasticSearch

What is ElasticSearch? Elasticsearch [17] is a distributed, open source search and analytics engine for textual, numerical, geospatial, structured, and unstructured data. It is built on Apache Lucene and represents the central component of the Elastic Stack, a set of open source tools for data ingestion, storage, analysis, and visualization, commonly referred to as the ELK Stack (after Elasticsearch, Logstash, and Kibana).

Why ElasticSearch?

- Fast Search. By default, Elasticsearch indexes all data in every field and each indexed field has a dedicated, optimized data structure. For example, text fields are stored in inverted indices, and numeric and geo fields are stored in BKD trees.
- Scalability of the search engine. The architecture underlying allows to grow from a small cluster to a large cluster automatically and with no particular issues.
- Document oriented (JSON). Elasticsearch uses JavaScript Object Notation, or JSON, as the serialization format for documents.
- Multilingual. The International Components for Unicode plugin is used to index and tokenize multilingual content. Based on character ranges, it decides whether to break on a space or character.
- Schema free. Documents can be indexed without explicitly specifying how to handle each of the different fields that might occur in a document. When dynamic mapping is enabled, Elasticsearch automatically detects and adds new fields to the index.

Back-end Components and Architecture.

[17] https://www.elastic.co/what-is/elasticsearch

- Node. A node is a single server which is a part of cluster, stores data and participates in the cluster's indexing and search capabilities.
- Cluster. A cluster is a collection of one or more nodes that together holds the entire data. It provides indexing and search capabilities across all nodes and is identified by a unique name.
- Index. An index is a collection of documents with similar characteristics and is identified by a name. It is the equivalent of a table in RDBMS.
- Document. A document is a basic unit of information which can be indexed. Each document is a collection of fields, which are the key-value pairs that contain the data. It is demonstrated in JSON and it is the equivalent of a row in RDBMS.
- Shards. A shard is a partition of data that is part of an index, and runs on a node.
- Replica shard. A replica is an exact copy of a primary shard that's typically placed on a node separate from the primary shard

From Document to Searchable Index: Analyzer. When a document is created in Elasticsearch, it goes through various phases. We are going to focus on how 'text' fields are analysed, and become eligible for full-text queries.

FIGURE 2.8. Creation of a New Document .

The analysis process id made of three sequential steps:

(1) The Character Filter is responsible for cleaning/reordering the strings before the tokenisation phase by adding, removing, or changing characters. An example of this could be to strip any HTML markup.
(2) The content passes to the Tokenizer which is responsible for dividing it into simple terms (tokens), which will usually be words. The 'standard' tokenizer basically splits

by whitespace and also removes most symbols, such as commas, periods, semicolons, etc. That's because most symbols are not useful when it comes to searching, as they are intended for being read by humans.

(3) Finally, it is run through zero or more token filters. A token filter may add, remove, or change tokens. Token filters are applied for operations such as removing stop words, converting to lowercase. A particularly interesting token filter is the one named 'synonym', which is useful for giving similar words the same meaning. For example, the words "nice" and "good" share the same semantics, although they are different words, and the same would be the case for the words "awful" and "terrible." So by using the synonym token filter, you could match documents containing the word "nice," even if you are searching for the word "good," because the meaning is the same, and therefore the document is highly likely to be as relevant as if the query used the other word.

The results of the analysis is actually what is stored within the index that a document is added to. More specifically, the analyzed terms are stored within an inverted index that we will describe further in the next paragraph.

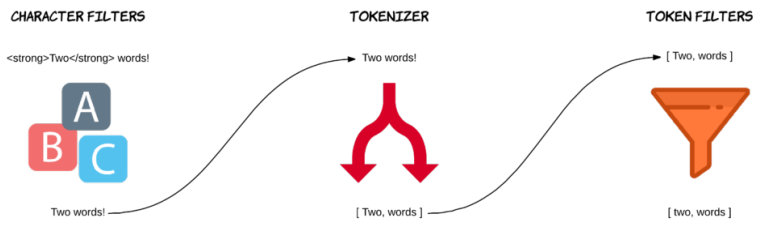


FIGURE 2.9. Standard Analyzer

Inverted Index. The inverted index stores text in a structure that allows for very efficient and fast full-text searches [52]. When performing full-text searches, we are actually querying an inverted index and not the JSON documents that we defined when indexing the documents. There will be an inverted index for each full-text field per index. So if an index contains documents that have five full-text fields, there will be five inverted indices.

An inverted index consists of all of the unique terms that appear in any document covered by the index. For each term, the list of documents in which the term appears, is stored. So essentially an inverted index is a mapping between terms and which documents contain those terms. Since an inverted index works at the document field level and stores the terms for a given field, it does not need to deal with different fields.

The inverted index also holds information that is used internally, such as for computing relevance. The relevance score is a strictly positive float that indicates how well each document satisfies the searching criteria [18].

Textual queries. Textual queries can be broken down into two families [19]:

- Term-based queries. They are low-level queries that have no analysis phase. A term query for the term 'Foo' looks for that exact term in the inverted index and calculates the relevance score for each document that contains the term. It won't match any variants like 'foo' or 'FOO'.
- The full-text query will use the same analyzer that was used while indexing the data. More precisely, the text of the query will go through the same transformations as the text data in the searching field.

Full-text queries. 'Match query' is the standard query for querying the text fields. The string that is passed into the query parameter, by default, going to be processed by the same analyzer as the one that has been applied to the searched field. Unless another analyzer is specified.

If instead of just a word, it is a phrase to be searched, it will be analyzed and the result will be a set of tokens. By default, Elasticsearch will be using OR operator between all of those tokens. That means that at least one should match, and more matches will hit a higher score. It might switched to AND. In this case, all of the tokens will have to be found in the document for it to be returned.

[18] https://www.elastic.co/guide/en/elasticsearch/guide/current/relevance-intro.html
[19] https://www.elastic.co/guide/en/elasticsearch/guide/current/term-vs-full-text.html

'Match phrase query' is the same as 'match' but the sequence order and proximity are important. Match query is not aware of the sequence and proximity.

Distributed Search. Elasticsearch enables scalability and availability thanks to the cooperation of its backend components listed above. Indexes are split into small parts called shards. At the index creation, Elasticsearch needs to know the number of shards your application wants for the index and Elasticsearch clusters will handle their management. As the system has more data, it can scale horizontally by adding more machines.

Let's assume that an index is dived into six shards and one replica, to partition across three nodes. It means the index will have 6 primary shards and 6 replica shards, a total of 12 shards. Since we have three nodes (servers) and twelve shards, each node will now contain four shards.

One of the reasons queries executed on Elasticsearch are so fast is because they are distributed. Multiple shards act as one index. A search query on an index is executed in parallel across all the shards. Any request can be sent to any node of the cluster as each one in the architecture is capable of handling any kind of request. In fact, all nodes know about all the other nodes in the cluster and can forward client requests to the appropriate node [20].

The node that receives the query assumes the role of coordinator, and broadcasts the query to each shard (both primary and replicated) of the index. Each shard performs the search and creates a local result list. Finally, all local results are delivered to the coordinator who merges them and makes a general result list and returns to the user in JSON format.

Failure handling. In a distributed environment, a node/server can go down due to various reasons, such as disk failure or network issue. To ensure availability, each shard, by default, is replicated to a node other than where the primary shard exists. If the node containing the primary shard goes down, the shard replica is promoted to primary, and the data is not lost, and the system can continue to operate on the index.

[20] https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html

Real Time. A real-time system can be classified as such if the delay between input and output meets a specified timing constraint.

Elasticsearch is the best solution known for real-time search because when a document is indexed, it will be eligible for search after one second.

How? As is known, the disks are usually a bottleneck for I/O operations. Also some mechanisms used for prevent loss of data and data persistence increases the cost in term of time.

ElasticSearch writes the new documents to the file-system cache first and only later it flushed to disk. A cached data can be read and opened like any other document without waiting for the commit on the disk. In Elasticsearch the shards are refreshed every second by default, and new documents are eligible for search after one second they have been indexed. This makes Elasticsearch architecture suitable for real-time search

2.2.3 Lean Software Development

Lean Software Development was originally formed as a methodology for manufacturing industry which originated the lean development process as a way to optimize production.

In fact, it was originally called the Toyota Production System, because Toyota invented this approach as a way to organise its production of cars, eliminating wasted time and resources.

Lean software development is an adaptation of lean manufacturing principles and practices to the software development domain.

There are 7 principles to Lean software development, each aiming to quicken delivery and bring higher value to end-user. Here we list some of them we found particularly interesting in our research and that lead us to use this methodology.

- Eliminating Waste, like unnecessary features and code, inefficient communication, vague requirements. Regular meetings are held after each short iteration to avoid bottlenecks and suggest which changes to implement during next iterations, which

facilitates learning and allows improvements to the code to be implemented in small, manageable increments.

- Delivering fast. Lean engineers came up with the concept of MVP (minimum viable product): build quickly, include little functionality and launch a product to the market as fast as possible. Then, study the reaction. Such approach allows to enhance a piece of software incrementally, based on the feedback collected from real customers, and ditch everything that is of no value. This is exactly the opposite philosophy of the Waterfall Methodology [21].
- Amplifying knowledge. Each time the code is written, engineers reflect on it immediately and then incorporate, during following iterations, the lessons they have learned. The clients get to voice feedback to the development team upon each iteration; collecting it and adjusting future efforts to the requirements is paramount to all lean developers.
- Building Quality In. Every small iteration, each loop is followed by an immediate assessment. The time between software development stages is always reduced as much as possible and trade-offs (occasional sacrifices of qualify for other project dimensions – time, costs and scope) are regularly discussed and considered.
- Delaying commitment. Many changes can emerge in technologies available and in market's course overall. Lean projects are bound to face uncertainty leaving room for improvement by postponing irreversible decisions until all the needed experimentation is done and as much info as possible is gathered.

In summary, the pillar of Lean Methodology is the Continuous Improvement of the software as shown in the figure below.

2.2.4 NLTK

NLP (Natural Language Processing) is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The purpose of the NLP is therefore to design the algorithms for the automatic processing of natural language.

[21] `https://en.wikipedia.org/wiki/Waterfall_model`

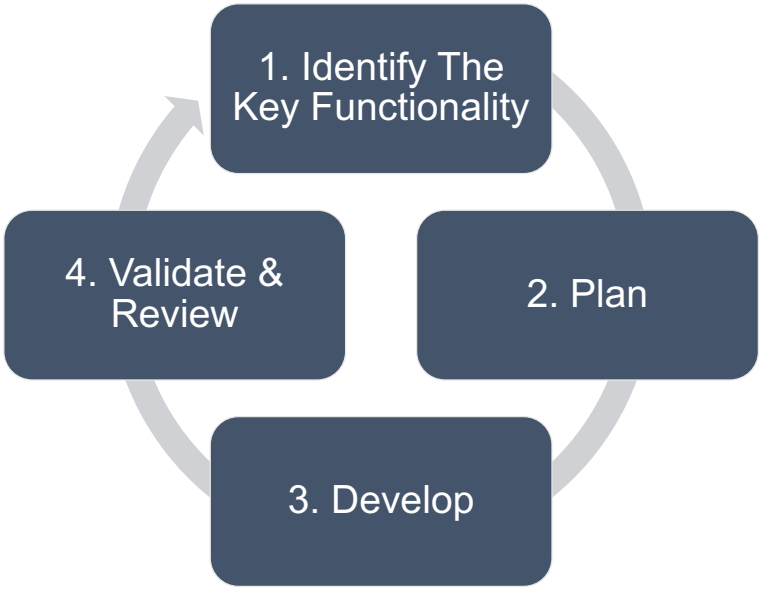


FIGURE 2.10. Continuous improvement is one of the Pillars of Lean, which guide all Lean methodology practice.

NLTK is one of the most used platform for building Python programs to work with human language data [22]. It provides a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Tokenization. Tokenization is the process of splitting a phrase or an entire text document into smaller units, such as individual words and each of these smaller units are called tokens. Tokenization is the base of NLP because analyzing the words that compose a docuement text is the first step to understand the meaning of the text itself.

Stop word removal. Stop-words are frequently words which do not add any semantic value to the textual context, such as articles or conjunctions and prepositions.

N-Grams. So far we have considered words as individual units, but the most interesting analysis actually need to consider the correlation between words. N-Grams are basically a

[22] http://www.nltk.org/

Tokenizing text

```
In [1]: text="Nowadays people know the price of everything and the value of nothing"
        from nltk.tokenize import word_tokenize, sent_tokenize

        sents = sent_tokenize(text)
        print(sents)

        ['Nowadays people know the price of everything and the value of nothing']

In [2]: words = [word_tokenize(sent) for sent in sents]
        print(words)

        [['Nowadays', 'people', 'know', 'the', 'price', 'of', 'everything', 'and', 'the', 'value', 'of', 'nothing']]
```

FIGURE 2.11. Tokenizing text in Jupyter Notebook.

Remove stop words

```
In [3]: from nltk.corpus import stopwords
        from string import punctuation
        customStopWords=set(stopwords.words('english') + list (punctuation)) # Set of stopwords in english + punctuation marks

In [4]: wordsWOStopwords=[word for word in word_tokenize(text) if word not in customStopWords]
        print(wordsWOStopwords)

        ['Nowadays', 'people', 'know', 'price', 'everything', 'value', 'nothing']
```

FIGURE 2.12. Removing stop words in Jupyter Notebook.

set of co-occuring words in a text within a given window. When N=1, this is referred to as unigrams and this is essentially the individual words in a sentence. When N=2, this is called bigrams and when N=3 this is called trigrams. When N>3 this is usually referred to as four grams or five grams and so on.

Let me explain with an example.

Unigram: [Let] [me] [explain] [with] [an] [example]

Bigram: [let me] [me explain] [explain with] [with an] [an example]

Trigram: [let me explain] [me explain with] [explain with an] [with an example]

Identify bigrams

```
In [7]: from nltk.collocations import *
        bigram_measures = nltk.collocations.BigramAssocMeasures()
        finder = BigramCollocationFinder.from_words(wordsWOStopwords) # Constructs bigrams from a list of words
        sorted(finder.ngram_fd.items()) # Print out ngram along with their frequencies

Out[7]: [(('Nowadays', 'people'), 1),
         (('everything', 'value'), 1),
         (('know', 'price'), 1),
         (('people', 'know'), 1),
         (('price', 'everything'), 1),
         (('value', 'nothing'), 1)]
```

FIGURE 2.13. Identifying bigrams in Jupyter Notebook

N-Grams can be used for many purposes. For example, they are used in Statistical Natural Language Processing for calculating the probability of the occurrence of a word after a certain word, and are the base for Sentence Completion systems.

Furthermore, they are also used in Text Summarization. In fact, by aggregating data at the n-gram level, it is possible to instantly pull out themes that would otherwise be difficult to identify when analyzing search terms in their entirety.

2.2.5 Jupyter Notebook

Jupyter Notebook is an open-source web application. The functionalities that it offers are (i) creating and (ii) sharing documents that contain live code, equations, visualizations and narrative text.

The Jupyter notebook combines two components. First, a web application for interactive authoring of documents with explanatory text, mathematics, computations and media output. Second, it provides notebook documents which are a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and media representations of objects.

Structure of a notebook document. The notebook consists of a sequence of cells. A cell is a multi line text input field and the execution behavior of a cell is determined by the cell's type. There are three types of cells: code cells, markdown cells, and raw cells.

A code cell allows the editing and writing of code, with full syntax highlighting and tab completion. The programming language depends on the kernel, and the default kernel runs Python code. When a code cell is executed, code that it contains is sent to the kernel, the results are retrieved and then displayed in the notebook. The output could be text, `matplotlib` figures and HTML tables (as used, for example, in the pandas data analysis package).

Markdown cells are used to document the computational process in a literate way through the Markdown language. It provides a simple way to perform the text markup, that allows the user to customize the text (e.g. emphasized (italics), bold, form lists, etc.)

Furthermore, it provides Markdown Headings, which consist of 1 to 6 hash # signs # followed by a space and the title of the section. It will be converted to a clickable link for a section of the notebook and also used as a hint when exporting to other document formats, like PDF.

CHAPTER 3

Methodology

This chapter describes the methodology used to interact with fact checkers and to develop the twitter monitoring tool. As said in the 1, our research is focused on the first step of the fact checking process which is the identification of check-worthy claims and topics. It is divided in two sections. The first one aims to further detail how the data have been collected from Twitter, what structure they have, the pre-processing made to clean the data and, finally, the indexing for allowing fast searches. The second section describes the *lean* methodology used to communicate with the experts, to develop the tool and its features, continuously improving and integrating the feedback obtained by the fact checkers.

3.1 Data Preparation

Before being available for analysis and be presented to fact checkers, data are crawled and pre-processed and finally indexed.

3.1.1 Data Crawling

Tweets are collected through Twitter's Sample Stream V1 API[1], using the `Twitter4j` client [2].

[1] `https://developer.twitter.com/en/docs/labs/sampled-stream/overview`

[2] `http://twitter4j.org/en/`

The data crawling start after sending a POST GET to the endpoint provided by the *Sample Stream v1* API [3]. It relies on an application-only authentication using OAuth 2.0 Bearer Token. Here is a an example of collected tweets. For simplicity we report just the more relevant fields.


```
{
"data": {
   "id": "1189226081406083073",
   "created_at": "2019-10-29T17:02:47.000Z",
   "text": "Sharing Tweets in DMs is our love language. Today, for
      Android users, we're making that easier.",
   "author_id": "783214",
   "in_reply_to_user_id": "783214",
   "referenced_tweets": [
     {
       "type": "replied_to",
       "id": "1151997885455581185"
     }
   ],
   "lang": "en",
   "source": "<a href="https://mobile.twitter.com"
      ......................
   }
}
```


To our purpose, the more relevant information are:

- *text* indicates the content of the tweet
- *created_at* is the time of creation of the tweet
- *id* is the unique number that identifies the user on Twitter
- *lang* is the language used in the tweet, automatically identified from the API

[3] https://api.twitter.com/labs/1/tweets/stream/sample

We chose Sampled Stream v1 API because it supports the collection of around 1% of publicly available tweets, as they happen. In the first stage of our research, the goal is to understand what are the relevant metadata collected and how to use them for Fact Checking purposes. We opted for an API that offers general and public information, instead of API that focuses on filtering of particular topics (e.g. Filtered stream v1) or on the retrieving of aggregated information, such as engagement metrics for any Tweet or list of Tweets from owned/authorized accounts [4].

3.1.2 Data Pre-Processing

Data are collected in JSON format. After the crawling, tweets are pre-processed with a Python script that filters tweets that have a non-empty text and English language.

```
def is_valid_tweet(doc):
    if "text" not in doc:
        return False
    if doc['lang'] == 'null' or doc['lang'] == 'None' or
        doc['lang'] != 'en':
        return False
    return True
```

3.1.3 Data Indexing

After crawling and pre-processing, we need to identify a technology that can store, search and and analyze big volumes of data quickly and in near real time. he data collection is indexed with ElasticSearch, a search engine built on Apache Lucene. It is a distributed, open source search and analytics engine for textual, numerical, geospatial, structured, and unstructured data. Every tweet represents a document in the index, every text field of the tweet is stored in an inverted index, which is a mapping between all the terms in the specific field and which documents contain those terms. There will be an inverted index for each full-text field of

[4] https://developer.twitter.com/en/docs/labs/tweet-metrics/overview

each tweet and it is a mechanism that allows for very efficient and fast full-text searches. For example, when we search for *coronavirus* in the field *text* of the entire collection of tweets, ElasticSearch will have already computed an inverted index for the field *text*, where it stores what are all the documents containing the keyword we selected.

Backend components:

- Index. The index is a collection of documents that have similar characteristics. An index is identified by a unique name that refers to the index when performing indexing search, update, and delete operations. In a single cluster, we can define as many indexes as we want. Index is the equivalent to database in an RDBMS.
- Document. A document is a basic unit of information that can be indexed. This document is expressed in JSON (JavaScript Object Notation). Analogy to a single raw in a DB. Within an index, you can store as many documents as you want, so that in the same index you can have a document for a single product, and yet another for a single order.

For further details on ElasticSearch, its basic components and features, see section 2.

Our approach. The indexing of the data happens in the beginning of each month. For example, the data collected in February will be available for analysis in March. The single tweet represents the smallest unit in the index, which is a document in ElasticSearch.

In our first approach, we index all the tweets, regardless of their creation time or topic, in one index. This immediately drove us in taking in consideration the following: Pros:

- It is a centralized solution, easily manageable.

Cons:

- Low performance. With the increasing of the data, performance in querying and searching are degraded, despite the natural predisposition to scale of ElasticSearch. Not compatible with a near real time requirement.

- Single point of failure

Given the previous considerations, we split the tweets in more than one index to achieve scalability and fast search queries. Tweets are divided by the date of its creation with a month granularity. The indexing on ElasticSearch every 5000 pre-processed documents, not document by documents.This reduces overhead and can greatly increase indexing speed. In order to know the optimal size of a bulk request, you should run a benchmark on a single node with a single shard. First try to index 100 documents at once, then 200, then 400, etc. doubling the number of documents in a bulk request in every benchmark run. When the indexing speed starts to plateau then you know you reached the optimal size of a bulk request for your data. Too large bulk requests might put the cluster under memory pressure when many of them are sent concurrently, so it is advisable to avoid going beyond a couple tens of megabytes per request even if larger requests seem to perform better. speed.

3.2 Lean Methodology

After the data crawling, pre-processing and indexing, the goal is to find the right methodology to interact with fact checkers and gather requirements from them that can help us in better developing a customized tool for their needs. It is important to highlight that the organizations has no Data Science and Software Engineering expertise, and our mission is to bridge for the first time the two sides. Our research is based on the Lean Methodology described in the chapter 2. Given the uncertainty that characterise the platform development and not knowing in advance what feature would be of interest and whatnot, we decided to proceed in an iterative way, following the principle of 'Continuous Improvement' and avoiding waste of time and resources, like developing features without previously consulting the experts and validate their usefulness.

Each iteration consists of four phases. In the first phase, the researchers –all of them with a Computer Science background– meet with the fact-checking experts –with a journalism and communication background– to identify key functionality, i.e., define what is the essential features that the tool must have.

In the second phase, researchers have brainstorming sessions to understand where the “low hanging fruit” is, and what is the most cost-effective way to apply text analytics and data visualization tools.

The third phase is the developing of the features towards a Minimal Viable Product (MVP) to show to the experts. This phase has a time span of three weeks, on average.

Finally, the MVP is validated by the experts, who provide feedback about the implemented features, which would also inform the starting of the next iteration.

CHAPTER 4

Results

This chapter follows the same structure of the Methodology chapter 3, providing practical results to what has been explained in theory so far. In fact, the first section provides details about the dataset that we have, following by a section where we further describe the results obtained in each iteration with the experts, analysing in detail each step of each iteration.

4.1 Dataset

We started collecting tweets from December 1, 2019, and as at May 1, 2020, the index contains a total of 182.1M tweets, with an average of 1.2M tweets per day.

Figure 4.1 gives an overview of how tweets are indexed in the ElasticSearch cluster. Each index has a storage size of 46 GB on average. Since the dimension of the indexes is always smaller than 50 GB, there is no need to divide the indexes into multiple shards. One replica is also created for handling faults and in the case the primary shard is not available.

Name ↑	Health	Status	Primaries	Replicas	Docs count	Storage size
2019-12	yellow	open	1	1	35154892	43.4gb
2020-01	yellow	open	1	1	37835603	51.8gb
2020-02	yellow	open	1	1	34042427	42gb
2020-03	yellow	open	1	1	36403562	45gb
2020-04	yellow	open	1	1	38670346	48gb

FIGURE 4.1. Kibana – Indexes overview

The pre-processing of the tweets reduces the initial dataset collected from the Twitter API of one third of the tweets actually crawled, as it is shown in the figure 4.2.

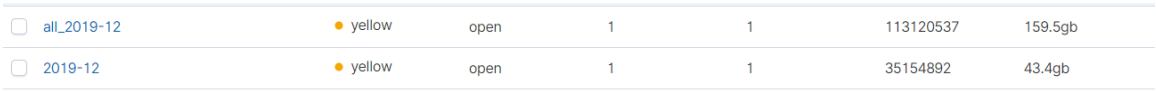


FIGURE 4.2. Kibana – Indexes comparison

4.2 Lean Methodology

The interaction with fact checkers and the development of *Watch n' Check* has followed an iterative methodology as described in the chapter 3. In this section we explain what are the results collected in each of the iteration, describing in details the connection between the proposed functionalities of the tool, the validation of the experts and the integration of their feedback in the development process.

4.2.1 First Iteration – Data presentation

The first question raised by the experts was about the access to public data published in social media platforms. Misinformation can be spread through multiple channels, including Facebook groups, Instagram conversations, or Twitter posts, among others. As described in the previous chapters, we have identified Twitter as our initial source to explore, as (i) it is one of the most important channels to spread information online, and (ii) it provides an API to extract a representative sample of the published information as it is released.

Proposed Functionalities. At this stage, the goal is to inform the fact checkers about the data and metadata of each tweet to understand what are the main relevant aspects that could support them in their fact checking process.

We used Kibana [1] as visualization tool, since it is part of the ELS Stack and offers visualization capabilities over an ElasticSearch cluster.

Furthermore, we planned and developed a console-based Python script that provided a simple but comprehensive way of inspecting the indexed collection of tweets, retrieving aggregate information.

[1] https://www.elastic.co/kibana

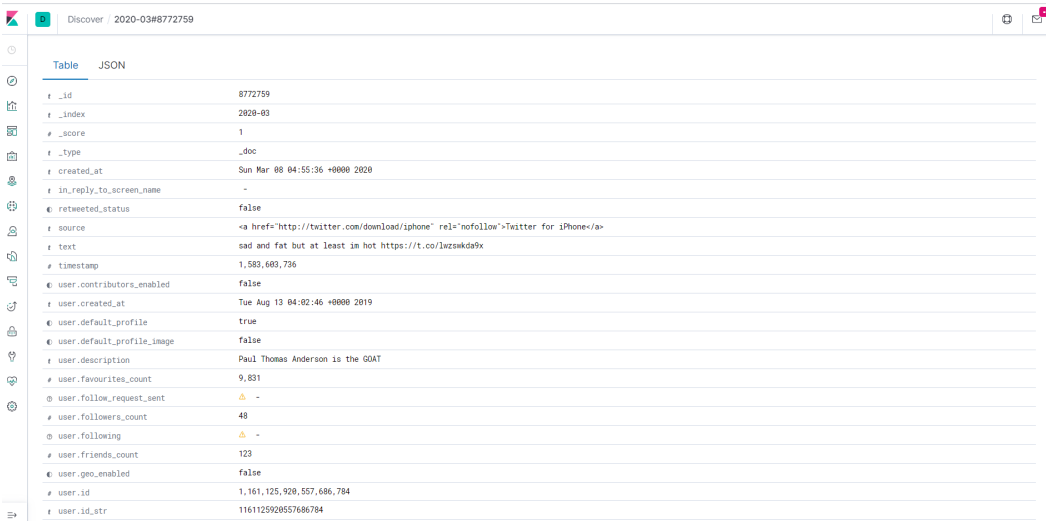


FIGURE 4.3. Kibana - Tweet fields

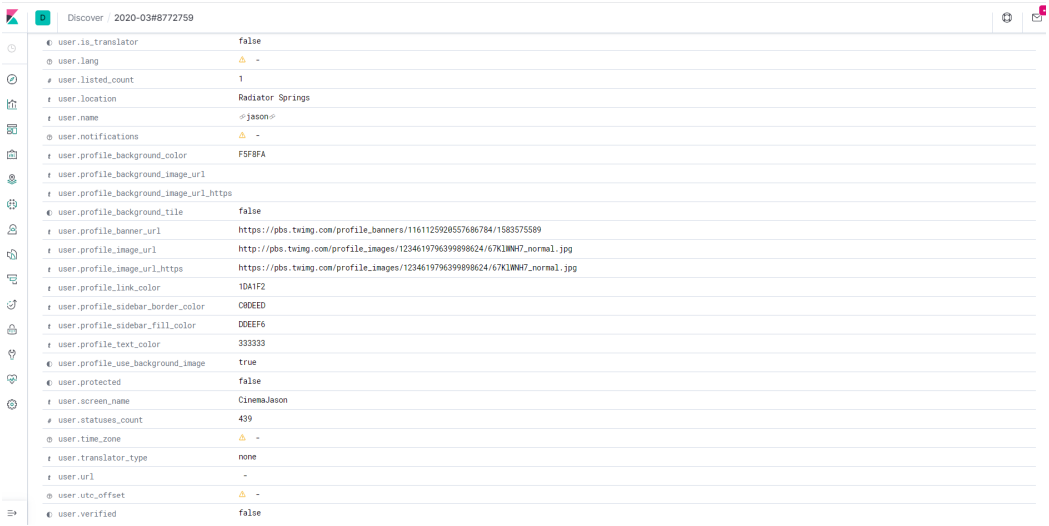


FIGURE 4.4. Kibana - Tweet fields - 2

This first proposed feature aimed to perform a quantitative real-time analysis on specific topics, filtering all the tweets with a 'text' field including a specified keyword. Here we report the query, see Full-Text search in Technologies section for more details.

```
def filter_keyword(esClient, keyword):
    search_body = {
        "query": {
            "match": {
                "text": {
```

```
                "query": keyword
            }
        }
    }
}
search(esClient, search_body)
```

Given a keyword, it computed and allowed the checking of:

- Number of tweets per month;
- Number of tweets per location;
- Number of tweets per user; and
- Number of tweets per day.

The first iteration aims to understand which data or metadata can provide more interesting and valuable insight for the fact checkers. Following the 'no time wasted' philosophy of the Lean Methodology, we did not focus, for example, on the aggregation of location. In fact, 'Melbourne' and 'Melbourne, Australia' appears as two different locations, while they are actually the same.

Validation & Review. At this stage, fact-checkers were invited to give their feedback on the current system in order to confirm its usefulness in their fact-checking process, and suggest new features. In this first phase, the following requirements were identified:

- Extend the analysis to a phrase or a set of words, instead of a single word. Fact-checkers were not only interested in the analysis of keywords (e.g., hashtags), but also in detecting and tracking entire statements or phrases (e.g., politicians' claims).
- Develop a user-friendly and compact visualization of the data. The experts would gain more insights with a graphical representations of the aggregated data (e.g., understand how the tweets about bushfires evolve over time in order to correlate this with external events).

- Provide access to tweet instances to have an understanding of their content. Besides descriptive statistics that summarize an aggregated sample of tweets, fact-checkers were interested in inspecting a sample of the textual content in the tweets, which could then potentially be used to perform more in-depth manual analysis.

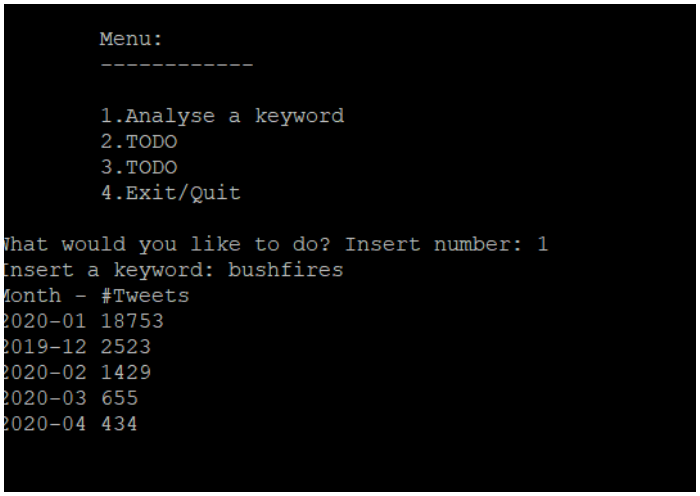


FIGURE 4.5. Python Console - Number of tweets per month about *bushfires*

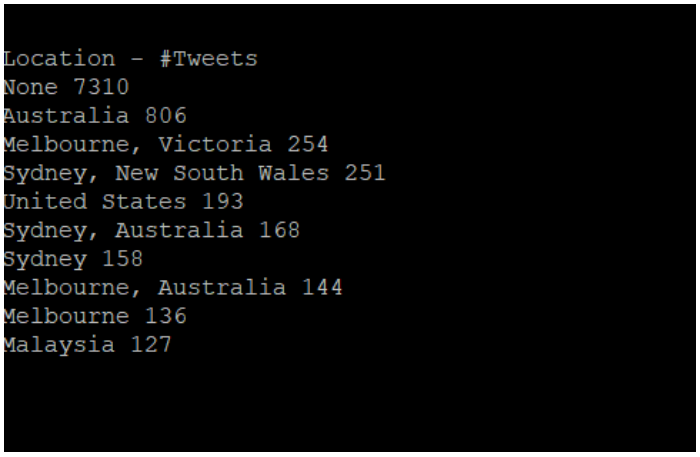


FIGURE 4.6. Python Console - Number of tweets per location about *bushfires*

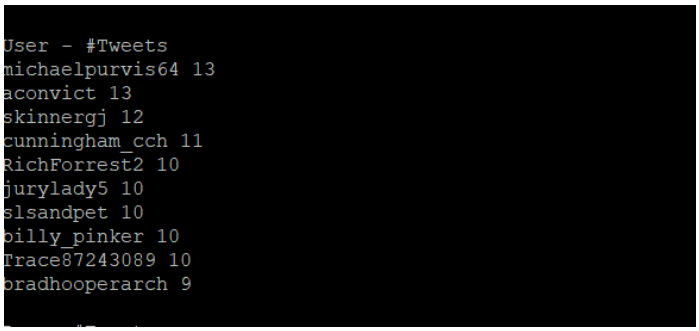


FIGURE 4.7. Python Console - Number of tweets per user about *bushfires*

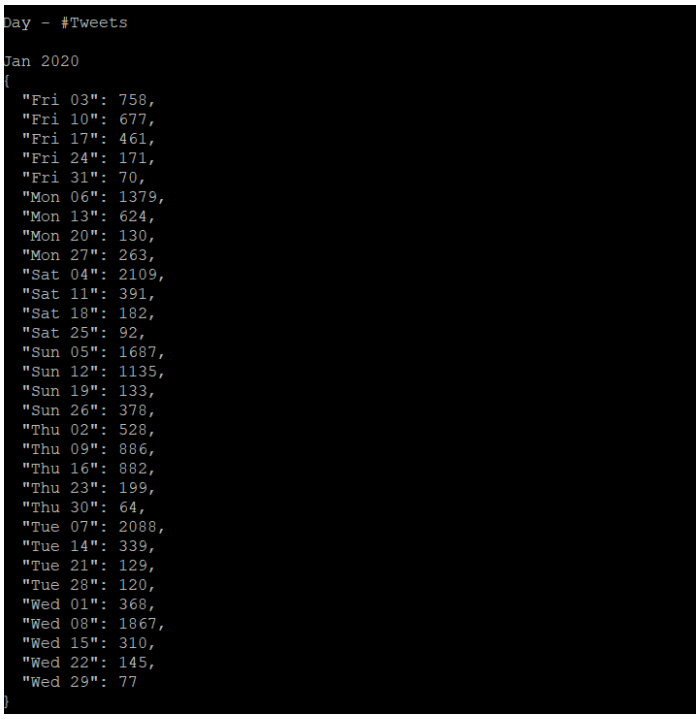


FIGURE 4.8. Python Console - Number of tweets per day about *bushfires*

4.2.2 Second Iteration – Visualizing Trends

The requirements identified in the previous iteration were used as a starting point for both the improvement of the current functionality, and the identification of new ones.

Therefore, we planned the following features to be developed during the second iteration:

- Visualization of the number of tweets containing a keyword over time.

- Computing the most frequent unigrams, bigrams, and trigrams as a mechanism to provide an overview over tweet texts.

Proposed Functionality. The first new functionality proposed is the visualization of the frequency of tweets related to a specific keyword/phrase over time. Figure 4.9, for instance, illustrates the frequency of tweets containing the keyword `bushfires` in our collection from December 2019 to April 2020.

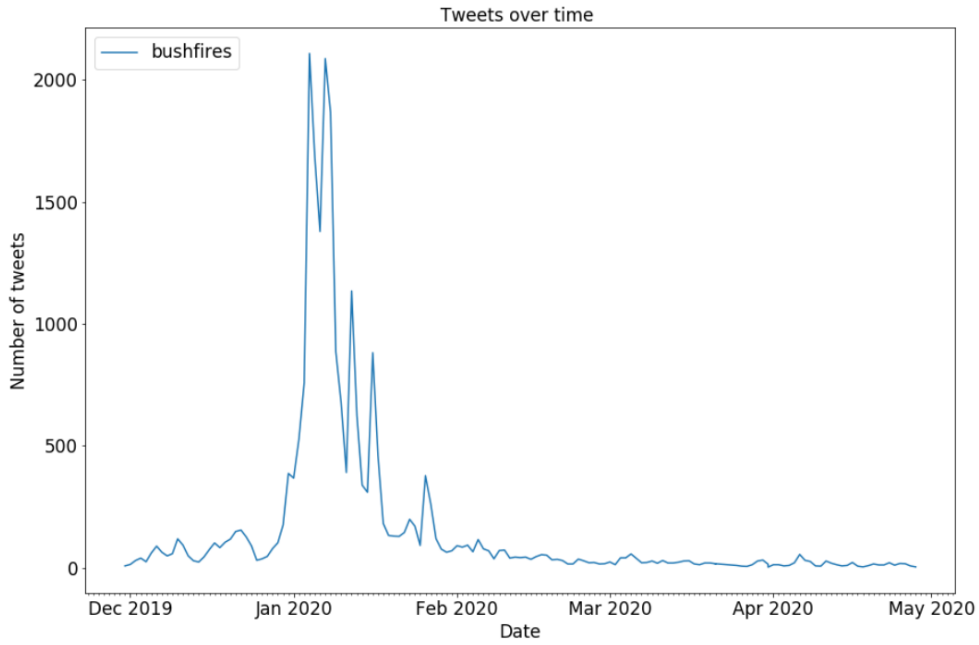


FIGURE 4.9. Frequency over time of tweets in the collection that contain the keyword `bushfires`.

The filtering of a phrase is done by using the ElasticSearch Full-text search as reported in Technologies section, using the following query:

```
def filter_phrase(client, phrase):
    search_body = {
        "query": {
            "match_phrase": {
                "text": phrase
            }
        }
```

```
}
search(client, search_body)
```

It enables us to search for a phrase, as shown in figure 4.10 for `climate change`.

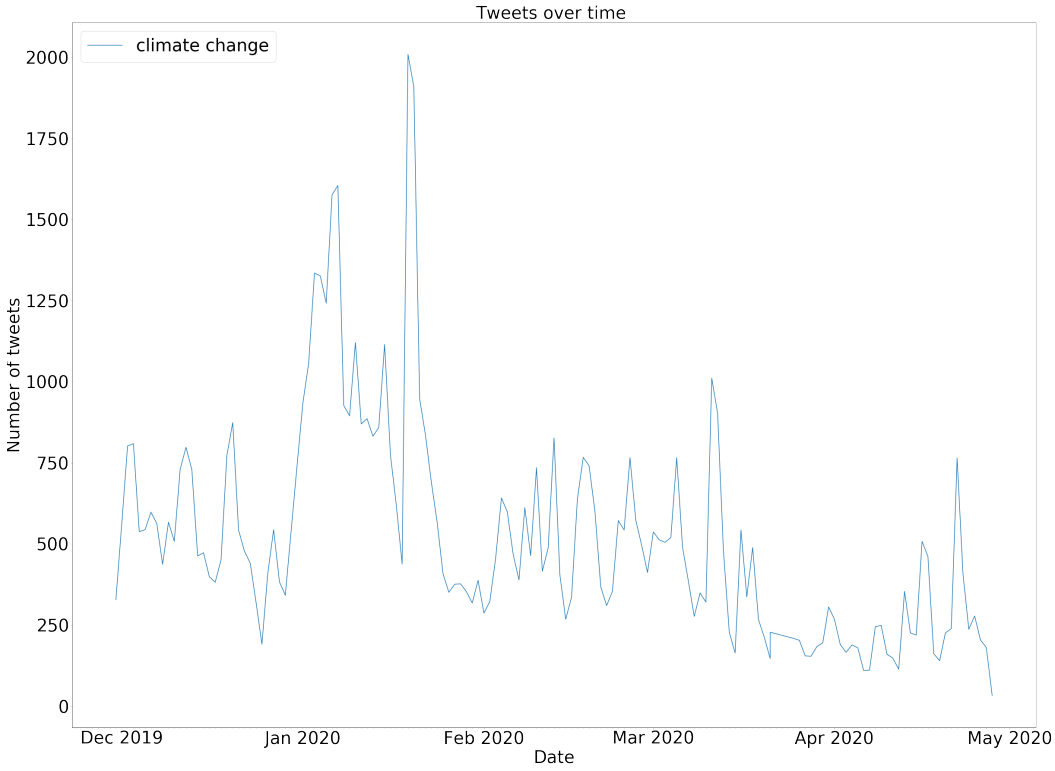


FIGURE 4.10. Frequency over time of tweets in the collection that contain the phrase `climate change`.

The second functionality aims to provide a text summarizing of the tweets in our collection, allowing to user to choose a period of time. This functionality is a direct consequence of the first one, since it allows the user to zoom on particular period of time showed in the previous graphs (e.g. peak time) and have an understanding of what the tweets are about. Using the NLTK libraries, we developed a simple mechanism to count the most co-occurrent unigrams, bigrams, and trigrams for a given keyword/phrase in the tweet texts. Before a tweet is made available for n-grams computing, it goes through a series of tasks that aim to delete information that is unnecessary for the research purposes and keep the fundamental information instead. This set of tasks include operations such as

(1) Tokenization
(2) Removing stop words
(3) Removing not alphabetical words

It also supports a dynamic choice of the period of time to analyze, and computes n-grams in that period. Figure 4.11 shows the graphs by *Watch n' Check* with the most co-occurrent n-grams for the keyword `bushfires` in the four indexed months of 2020.

Some questions n-grams can help answer are:

- What topics are the tweets related to in this particular period?
- What are the most occurent words in the tweets?
- How the language used by politicians, news, journals, is influencing tweets?
- How can we possibly change the language we used to communicate and integrate the language used on social networks?

Validation & Review. The first feature of this second iteration allows the experts to verify how a certain keyword evolves over time on Twitter. They also found it interesting to understand how a trend for a given keyword compares with the trend for other keywords/phrases. This can guide experts *a priori* in focusing their attention on the analysis of some topics instead of others, or it can be used as a tool to retrospectively validate the choice to focus on fact-checking certain content.

The n-gram graphs provide an approximate overview of what the topical content of the tweets. It gives also the opportunity to find new keywords to analyse. For example, in the figure 4.11, the word `princessaspien` occurs frequently. The experts would then research about it and its correlation with bushfires. They would also use the *Watch n' Check* tool to analyse the keyword `princessaspien` to see what is its trend over time in the tweets and n-grams related to it.

While it is useful to have a general idea, the fact-checking experts need to have a more detailed introspection on the content of the tweets, to check their correlation with some news (or fake

news), and to potentially track their spread. Another aim of the fact-checkers is to gain an understanding about the popularity of some claims or user profiles rather than keywords.

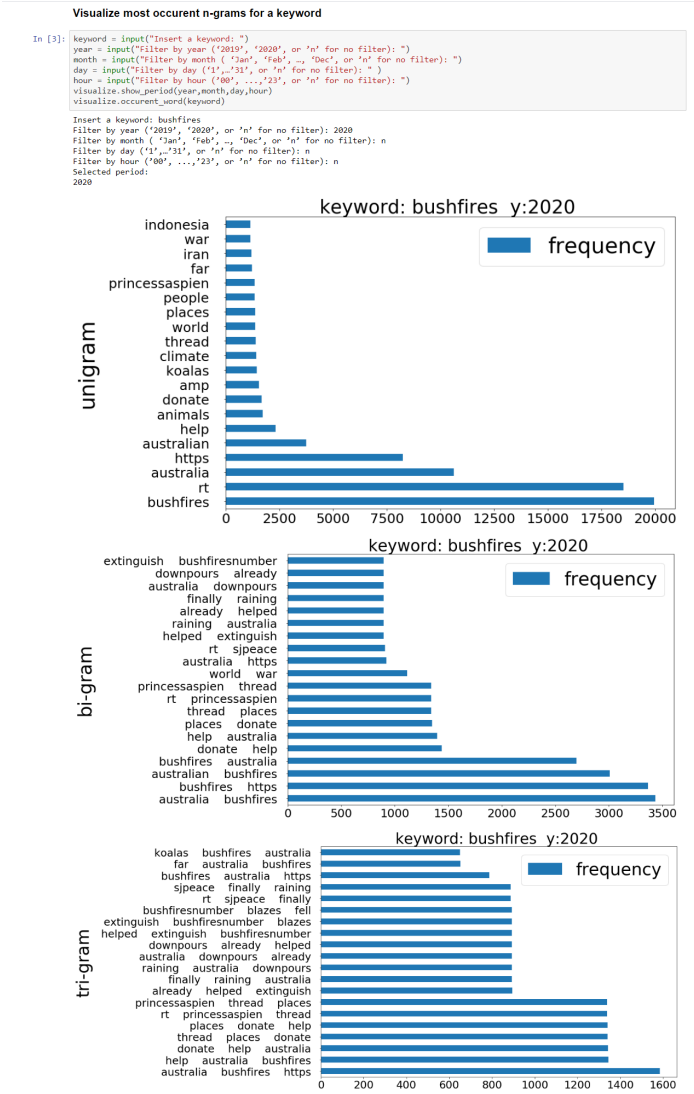


FIGURE 4.11. Frequency of n-grams which co-occur with the keyword `bushfires`.

4.2.3 Third Iteration – Towards a User-friendly Interface

After two iterations using a console-based interface operated by the researchers, Jupyter Notebook[2] has been identified as a tool that would allow to quickly provide a more friendly user interface.

Another key functionality that was identified is the relative comparison of multiple keywords over time, as it would provide an easy and immediate way to analyze the lifespan of different keywords in a single visualization.

Proposed Functionality. In order to provide a user interface that would allow experts to generate their own analyses, we opted to use Jupyter Notebook. The notebook extends the console-based approach to a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. In fact, it provides the possibility to run code real-time in specific code cells. Plus, it is possible to organize the notebook as a document, specifying titles of the different sections with a Markup Language in Markdown cells. More details about the Notebook and its functions are available in the section Background.

Although not ideal – a web front-end is planned for future work as described in Section 6 – it provided a flexible and quick way to show results in a more friendly manner.The new interface allows access to all the functionality from a browser with access to the intranet.

The second new functionality of the the tool provides a comparison of the trends of multiple keywords simultaneously. Figure 4.12 illustrates this functionality for the keywords `bushfires`, `climate change`, `coronavirus`, and `vaccine`. As different keywords may generate curves with substantial changes in the range of frequencies, a graph with a logarithmic scale is also generated. A trend is the general direction that a particular topic is taking during a specified period of time. Trend analysis is the process of trying to look at current trends in order to understand how it evolves compared to others. In the fact checking process it can be used as a tool to understand what is the up-trending topic, and consequently

[2] `https://jupyter.org/`

focus the fact checking on that specific one. In fact, the more spread is a topic, the more likely are the fake-news related to it. For example, in the case of Figure 4.12, we can clearly notice that the number of tweets that contains the keyword *coronavirus* is much higher than the others. Although it is obvious and expected in this case, it can be more informative in other cases where the extent of the topics is not so different, like in Figure 4.13

Validation & Review. The comparison of the frequencies over time provides valuable insights for the experts. In fact, one of the graphs generated by the tool has been included in their weekly fact-checking newsletter. An interesting observation at this stage was that the linear scale version of the graph was preferred over the logarithmic scale counterpart. This suggests that, although some visualization techniques may provide a clearer representation of skewed data, these may be harder to interpret.

The functionalities of *Watch n' Check* integrated into a Jupyter Notebook seem to appear clear to experts. However, we acknowledge that is not optimal, as the user could inadvertently change the structure of the code and compromise the analysis itself.

Furthermore, the fact checking process analyzes the truthfulness of news and claims on a daily/weekly basis. This highlights the need to have a real-time application and create a user interface easily accessible by fact-checkers.

Compare time graphs of different keywords

```
In [4]: visualize.combined_tweets_per_day("","more_keywords")
        visualize.log_combined_tweets_per_day("","more_keywords")
```

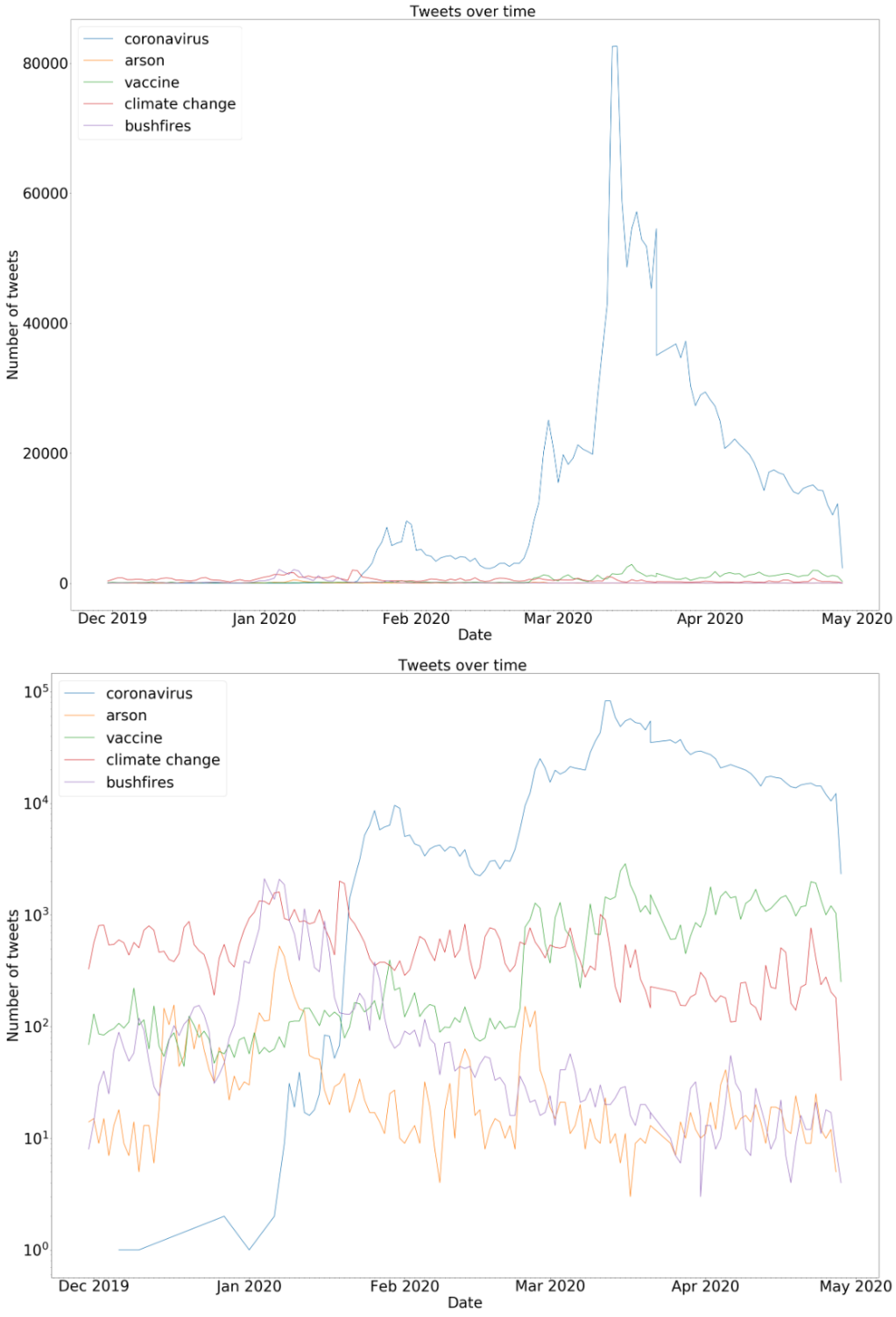


FIGURE 4.12. Comparison of the frequency over time of tweets in the collection that contains the specified keywords.

Compare time graphs of different keywords

```
In [5]: visualize.combined_tweets_per_day("","more_keywords")
        visualize.log_combined_tweets_per_day("","more_keywords")
```

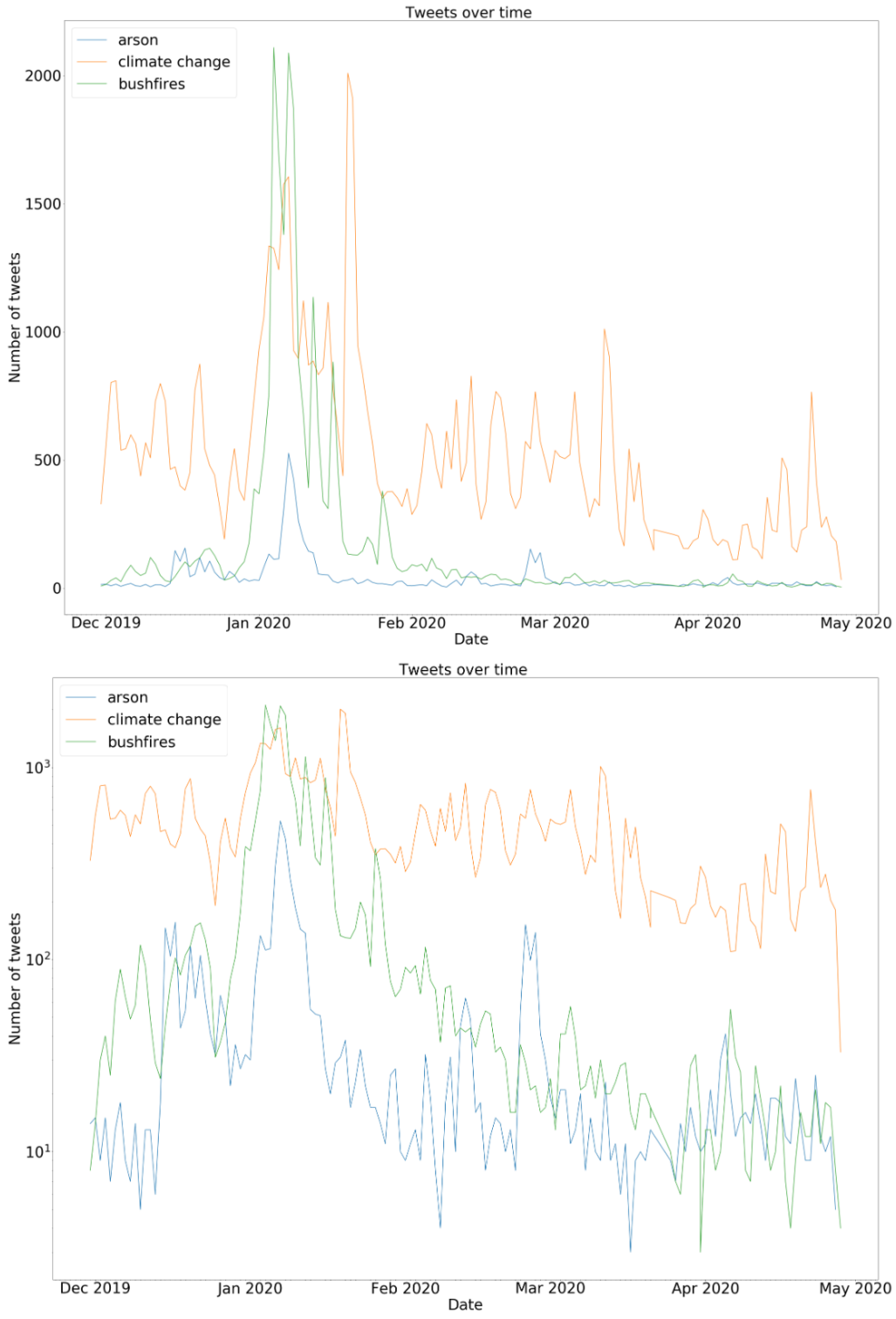


FIGURE 4.13. Comparison of the frequency over time of tweets in the collection that contains the specified keywords.

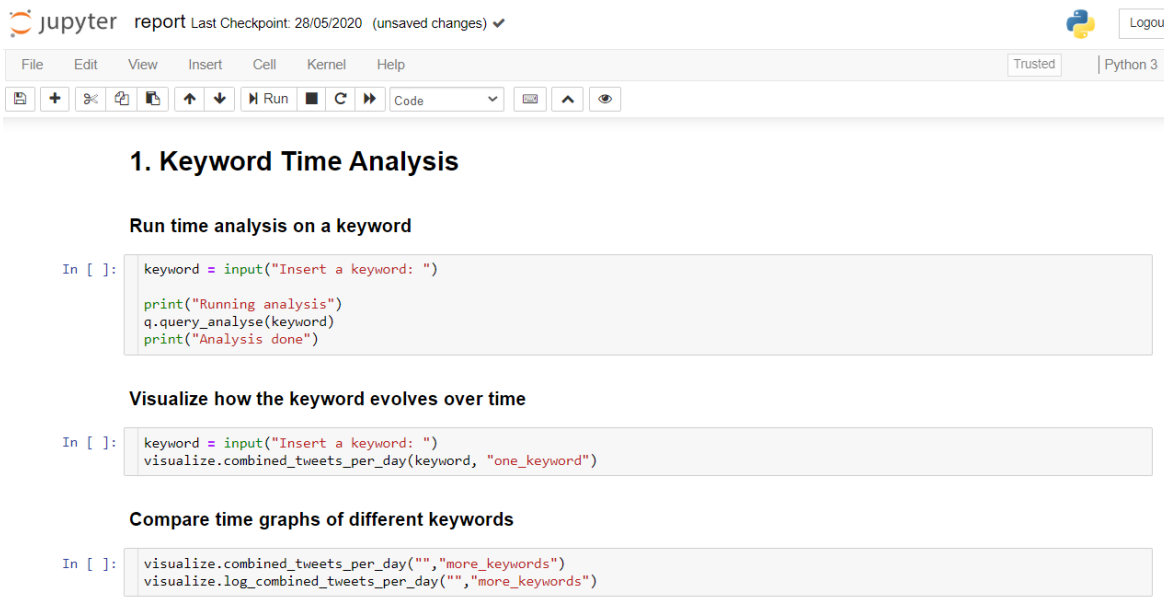


FIGURE 4.14. Jupyter Notebook - First Section

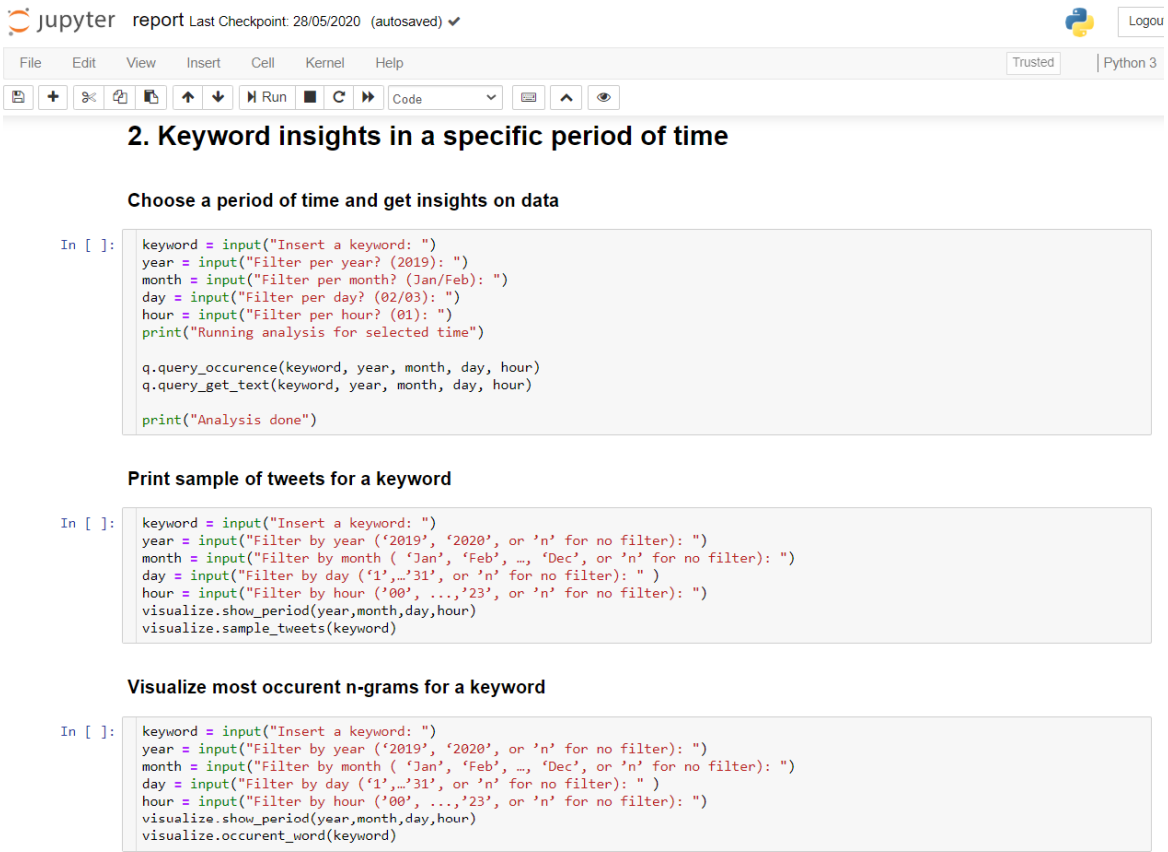


FIGURE 4.15. Jupyter Notebook - Second section

CHAPTER 5

Discussion

Fact checking organizations have proved to be a key player in the fight against the misinformation. On the other side, social networks have facilitated the spread of fake news due to their rapidity in the dissemination of contents.[14, 4, 38, 9] Although the fact checking outlets are already aware of the power of the social media and the use that some politicians and figures of influence do [2], the methodology that they currently use in their daily work is strongly relying on (i) the traditional channels of communication (e.g. TV, radio, etc.) [47, 7] and (ii) on automatic tools that provide results computed with advanced mechanisms [10], such as machine learning or natural language processing, that appear as a black box where the fact checkers have no transparency. Our research has recognized these two points as the main contributors that guided us towards the answer to the research questions we identified in the beginning of the journey.

5.1 Findings

First, we validated with the experts that Twitter is a valuable source for more advanced data analysis, since it is one of the main platform for political discussion and have a broad reach of people, and potentially used from politicians to claim statements. The amount of data generated is huge and not easily manageable without a proper software architecture. Although the increasing number of social media monitoring systems on the market, there are no specific tools for fact checking. The majority, in fact, focuses on marketing analysis, etc. They present common patterns that we have reproduced in our final version of *Watch n' Check*: an architecture for data tracking, preparation, analysis and presentation of information [43, 13],

plus access to historical data, near-real time analysis and a dashboard that offers graphical representation of the raw data.

Our research shows that fact checking organizations are mainly composed by journalists and other experts, with a very low percentage of computer science expertise in the team. [7]The organizations across the world also differ in their internal methodology. This has proved the need for a customized tool, as *Watch n' Check* aims to be. We find that an iterative approach can help to develop a tool that (i) satisfies the particular needs of the specific organization and (ii) engages the experts in the process of development and understanding of the analysis in a cost-effective way. In fact, the iteration of *propose feature - experts feedback and validation - developing* has driven us to valuable findings - both for us and for the experts.

We found that providing mechanisms to track keywords/phrases over time in social networks can give them a valuable idea of how certain topics evolve compared to others. This can guide experts *a priori* in focusing their attention in analysing some topics instead of others, or it can be used as a tool to retrospectively validate the choice to focus on fact-checking certain content. The more spread is a topic on social networks, the more likely are the fake-news related to it. Although the influence and the reach that some topics have is obvious and expected in some cases, it could be less trivial in others. Trend-comparison is an valuable informative tool to have in the latter case.

Finally, most co-occurrent n-grams are as important as providing access to instances of data (i.e., retrieving tweets containing the keywords of interest) because they provide a summarize over a specified period of time, thus an overview of what the tweets are about. This also leads to find other keywords strictly related to the searched topic that potentially lead to other claim in tweets.

5.2 Limitations and Further Development

The current version of *Watch n' Check* includes several limitations and avenues for further development.

First, the current prototype relies on the sample of tweets collected via the Twitter Stream API – which exposes about 1 percent of publicly available tweets – and is further filtered by language. Therefore, *Watch n' Check* can be used as a complementary tool to help identify relevant information, but experts will still need to access the original platform to refine their analyses. A future expansion could use *Watch n' Check* in the first stage as a tool to identify the most spread topics on Twitter, and then retrieve specific data about that specific topics using the Filter real-time tweets API. The amount of data, thus the analysis that can be done, could increase with the access to multiple social media platforms such Instagram and Facebook. However, they have more restrictive access to public data.

The platform is only analysing English tweets. A broader and more detailed analysis would include tweets in different languages.

Watch n' Check filters tweets by matching the specified keyword or phrase. However, a semantic representation of the *topic* would enhance the analysis. To this aim, we plan to incorporate topic models [5], so if the experts filter tweets about `coronavirus`, the tool would also filter tweets that are semantically related (e.g., tweets that include `covid`). A further development of *Watch n' Check* could include more advanced technologies for text summarizing [27, 24], to get a more sophisticated overview of the happenings related to a topic/event.

The current user interface provided through Jupyter Notebook was used for the purpose of offering a Minimal Viable Product, as an easy way to let the experts validate the functionalities provided. The development of a web front-end is part of our immediate future work.

Watch n' Check represents the base for more advanced and accurate analysis that can improve the connection between fact-checking and information propagation in social media, such as topic modeling, and semantic and sentiment analysis for specific arguments. Moreover, such analysis can be extended beyond the information content by analyzing the social network structure and the users who engage with this content. In this way, the fact-checkers can effectively analyze trends and communities and their associations to make better decisions when validating information in social media.

CHAPTER 6

Conclusion

Fake news are an increasing phenomenon and their broad reach on social networks represents a threat in the global world. Fact checking organizations play a key role in the fight against the fake news spread, and providing them a systematic and programmatic access to the huge amount of data produced every day is a necessity now more than ever. Our collaboration with RMIT ABC Fact Checking has proved that the use of pre-fabricated and automated tool for claim detection can bring valuable results, but in the same time appear as a black box which does not provide any transparency on the way that results are elaborated. Thus, our work address the need to create in a cost effective way a tool that support fact checkers in the identification of check worthy topics/claims, instead of replacing them. Supporting them also means that they have control on the output of the analysis, which requires them being part of the whole process of developing. We found a lack of literature about the engagement of fact checkers in the automated tools development currently on the market, which drove us in pursuing this research. (i)Understanding what are the most valuable aspects of the social data and (ii) performing data analysis on them, with an iterative evaluation and integration of experts feedback is been the unique value of our work. The result is a twitter monitoring tool that we named *Watch n' Check* that aims to add value in the organization and drive experts in a faster identification of check worthy claims/topics on Twitter. An open challenge that remains is to understand how those automated tools can be actually integrated in the fact-checking process in an effective way, to support the experts in their daily work. One other challenge in designing information access tools for fact-checkers – and for any given group of experts in general – consists of having a clear way to explain the output of the system, e.g., how a conclusion is made and with how much confidence, to ensure the main objective of

such system, i.e., empowering experts through providing timely and accurate information, is reached.

Bibliography

[1] Firoj Alam, Shaden Shaar, Fahim Dalvi, Hassan Sajjad, Alex Nikolov, Hamdy Mubarak, Giovanni Da San Martino, Ahmed Abdelali, Nadir Durrani, Kareem Darwish and Preslav Nakov. *Fighting the COVID-19 Infodemic: Modeling the Perspective of Journalists, Fact-Checkers, Social Media Platforms, Policy Makers, and the Society*. 2020. arXiv: 2005.00033 [cs.CL].

[2] Hunt Allcott and Matthew Gentzkow. 'Social Media and Fake News in the 2016 Election'. In: *Journal of Economic Perspectives* 31.2 (2017), pp. 211–36. DOI: 10.1257/jep.31.2.211. URL: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211.

[3] Pepa Atanasova, Alberto Barron-Cedeno, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Martino and Preslav Nakov. 'Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. Task 1: Check-Worthiness'. In: *Proceedings of CLEF'18*. Aug. 2018.

[4] Alessandro Bessi and Emilio Ferrara. 'Social bots distort the 2016 U.S. Presidential election online discussion'. In: *First Monday* 21 (Nov. 2016). DOI: 10.5210/fm.v21i11.7090.

[5] David M Blei. 'Probabilistic Topic Models'. In: *Communications of the ACM* 55.4 (2012), pp. 77–84.

[6] RMIT ABC Fact Check. *No, wearing a face mask doesn't cause carbon dioxide toxicity*. 2020.

[7] Federica Cherubini and Lucas Graves. 'The rise of fact-checking sites in Europe'. In: *Reuters Institute for the Study of Journalism, University of Oxford. http://reutersinsfitute. polifics. ox. ac. uk/our-research/rise-fact-checking-sites-europe* (2016).

[8] David Child. *Fighting fake news: The new front in the coronavirus battle*. 2020.

[9] Z. Chu, S. Gianvecchio, H. Wang and S. Jajodia. 'Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?' In: *IEEE Transactions on Dependable and Secure Computing* 9.6 (2012), pp. 811–824.

[10] Nadia K. Conroy, Victoria L. Rubin and Yimin Chen. 'Automatic deception detection: Methods for finding fake news'. In: *Proceedings of the Association for Information Science and Technology* 52.1 (2015), pp. 1–4. DOI: 10.1002/pra2.2015.145052010082. eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/pra2.2015.145052010082. URL: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/pra2.2015.145052010082.

[11] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H. Eugene Stanley and Walter Quattrociocchi. 'The spreading of misinformation online'. In: *Proceedings of the National Academy of Sciences* 113.3 (2016), pp. 554–559. ISSN: 0027-8424. DOI: 10.1073/pnas.1517441113. eprint: https://www.pnas.org/content/113/3/554.full.pdf. URL: https://www.pnas.org/content/113/3/554.

[12] Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Martino and Pepa Atanasova. 'CheckThat! at CLEF 2019: Automatic Identification and Verification of Claims'. In: Apr. 2019, pp. 309–315. ISBN: 978-3-030-15718-0. DOI: 10.1007/978-3-030-15719-7_41.

[13] Weiguo Fan and Michael Gordon. 'The Power of Social Media Analytics'. In: *Communications of the ACM* 57 (June 2014), pp. 74–81. DOI: 10.1145/2602574.

[14] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer and Alessandro Flammini. 'The rise of social bots'. In: *Communications of the ACM* 59.7 (2016), 96–104. ISSN: 1557-7317. DOI: 10.1145/2818717. URL: http://dx.doi.org/10.1145/2818717.

[15] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. 2000.

[16] Caroline Fisher, Sora Park, Jee Young Lee, Glen Fuller and Yoonmo Sang. *Digital News Report: Australia 2019*. Tech. rep. University of Canberra, 2019.

[17] Eibe Frank and Mark Hall. 'A Simple Approach to Ordinal Classification'. In: *Machine Learning: ECML 2001*. Ed. by Luc De Raedt and Peter Flach. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 145–156. ISBN: 978-3-540-44795-5.

[18] Pepa Gencheva, Ivan Koychev, Lluís Màrquez, Alberto Barrón-Cedeño and Preslav Nakov. *A Context-Aware Approach for Detecting Check-Worthy Claims in Political Debates*. 2019. arXiv: 1912.08084 [cs.CL].

[19] Lucas Graves. *Understanding the Promise and Limits of Automated Fact-Checking*. Tech. rep. Reuters Institute, University of Oxford, 2018.

[20] Lucas Graves and Michelle A. Amazeen. 'Fact-Checking as Idea and Practice in Journalism'. In: 2019.

[21] Naeemul Hassan, Bill Adair, James Hamilton, Chengkai Li, Mark Tremayne, Jun Yang and Cong Yu. 'The Quest to Automate Fact-Checking'. In: *Proceedings of the 2015 Computation + Journalism Symposium* (Oct. 2015).

[22] Naeemul Hassan, Chengkai Li and Mark Tremayne. 'Detecting Check-worthy Factual Claims in Presidential Debates'. In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM)* (Oct. 2015), pp. 1835–1838. DOI: 10.1145/2806416.2806652.

[23] Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li and Mark Tremayne. 'ClaimBuster: The First-ever End-to-end Fact-checking System'. In: *Proc. VLDB Endow.* 10 (2017), pp. 1945–1948.

[24] David Inouye and Jugal Kalita. 'Comparing Twitter Summarization Algorithms for Multiple Post Summaries'. In: Oct. 2011, pp. 298–306. DOI: 10.1109/PASSAT/SocialCom.2011.31.

[25] Andreas Jungherr. 'Twitter use in election campaigns: A systematic literature review'. In: *Journal of Information Technology & Politics* 13.1 (2016), pp. 72–91. DOI: 10.1080/19331681.2015.1132401. eprint: https://doi.org/10.1080/19331681.2015.1132401. URL: https://doi.org/10.1080/19331681.2015.1132401.

[26] Harriet Kasper, Moritz Dausinger, Holger Kett and Thomas Renner. *Social media monitoring Tools - IT-Lösungen zur Beobachtung und Analyse Unternehmensstrategisch relevanter Informationen im Internet*. Fraunhofer Verlag, Stuttgart, 2010.

[27] M. A. H. Khan, D. Bollegala, G. Liu and K. Sezaki. 'Multi-tweet Summarization of Real-Time Events'. In: *2013 International Conference on Social Computing*. 2013, pp. 128–133.

[28] Alexios Mantzarlis. *The 'Holy Grail' of Computational Fact Checking – and What We Can Do in the Meantime*. 2015. URL: https://www.poynter.org/fact-checking/2015/the-holy-grail-of-computational-fact-checking-and-what-we-can-do-in-the-meantime/.

[29] Amy Mitchell, Katie Simmons, Katerina Eva Matsa, Laura Silver, Elisa Shearer, Courtney Johnson, Mason Walker and Kyle Taylor. *In Western Europe, Public Attitudes Toward News Media More Divided by Populist Views than Left-Right Ideology*. 2018.

[30] Joe Murphy, Annice E Kim, Heather Hagood, Ashley Richards, Cynthia Augustine, Larry Kroutil and Adam J Sage. 'Twitter Feeds and Google Search Query Surveillance : Can They Supplement Survey Data Collection ?' In: 2011.

[31] Eni Mustafaraj and Panagiotis Takis Metaxas. *The Fake News Spreading Plague: Was it Preventable?* 2017. arXiv: 1703.06988 [cs.SI].

[32] ABC News. *Hundreds die in Iran over false belief drinking methanol cures coronavirus*. 2020.

[33] Raymond Nickerson. 'Confirmation Bias: A Ubiquitous Phenomenon in Many Guises'. In: *Review of General Psychology* 2 (June 1998), pp. 175–220. DOI: 10.1037/1089-2680.2.2.175.

[34] Anne Osterriede. 'The Value and Use of Social Media as Communication Tool in the Plant Sciences'. In: *Plant Methods* 9.13 (2013).

[35] John Parmelee. 'The Agenda-Building Function of Political Tweets'. In: *New Media & Society* 16 (Apr. 2013), pp. 434–450. DOI: 10.1177/1461444813487955.

[36] Paul, Christopher and Miriam Matthews. 'The Russian "Firehose of Falsehood" Propaganda Model: Why It Might Work and Options to Counter It'. In: *RAND Corporation* (2016). DOI: https://www.rand.org/pubs/perspectives/PE198.html.

[37] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff and Benno Stein. 'A Stylometric Inquiry into Hyperpartisan and Fake News'. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 231–240. DOI: 10.18653/v1/P18-1022. URL: https://www.aclweb.org/anthology/P18-1022.

[38] Quattrociocchi, Walter, Scala, Antonio, Sunstein and Cass R. *Echo Chambers on Facebook*. June 2016. DOI: http://dx.doi.org/10.2139/ssrn.2795110.

[39] Lee Ross and Andrew Ward. 'Naive realism in everyday life: Implications for social conflict and misunderstanding.' In: 1996.

[40] Paul Sakkal. ''Go back to your country': Chinese international students bashed in CBD'. In: *The Age* (2020).

[41] Elisa Shearer and Jeffrey Gottfried. *News Use Across Social Media Platforms 2017*. 2017.

[42] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang and Huan Liu. *Fake News Detection on Social Media: A Data Mining Perspective*. 2017. arXiv: 1708.01967 [cs.SI].

[43] Stefan Stieglitz, Linh Dang-Xuan, Axel Bruns and Christoph Neuberger. 'Social Media Analytics: An Interdisciplinary Approach and Its Implications for Information Systems'. In: *Business & Information Systems Engineering* Forthcoming (Jan. 2014). DOI: 10.1007/s11576-014-0407-5.

[44] Sebastian Stier, Arnim Bleier, Haiko Lietz and Markus Strohmaier. 'Election Campaigning on Social Media: Politicians, Audiences, and the Mediation of Political Communication on Facebook and Twitter'. In: *Political Communication* 35.1 (2018), pp. 50–74. DOI: 10.1080/10584609.2017.1334728. eprint: https://doi.org/10.1080/10584609.2017.1334728. URL: https://doi.org/10.1080/10584609.2017.1334728.

[45] *TV Main Source of News – and Most Trusted*. Data retrieved from Roy Morgan Single Source July 2017 - June 2018, http://www.roymorgan.com/findings/7746-main-sources-news-trust-june-2018-201810120540. 2018.

[46] Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño and Preslav Nakov. *It Takes Nine to Smell a Rat: Neural Multi-Task Learning for Check-Worthiness Prediction*. 2019. arXiv: 1908.07912 [cs.CL].

[47] Andreas Vlachos and Sebastian Riedel. 'Fact Checking: Task definition and dataset construction'. In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA: Association for Computational Linguistics, June 2014, pp. 18–22. DOI: 10.3115/v1/W14-2508. URL: https://www.aclweb.org/anthology/W14-2508.

[48] *WWW '11: Proceedings of the 20th International Conference on World Wide Web*. Hyderabad, India: Association for Computing Machinery, 2011. ISBN: 9781450306324.

[49] Reza Zafarani, Xinyi Zhou, Kai Shu and Huan Liu. 'Fake News Research: Theories, Detection Strategies, and Open Problems'. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, 3207–3208. ISBN: 9781450362016. DOI: 10.1145/3292500.3332287. URL: https://doi.org/10.1145/3292500.3332287.

[50] R. B. Zajonc. 'Attitudinal effects of mere exposure'. In: *Journal of Personality and Social Psychology* 9 (1968), pp. 1–27. DOI: https://doi.org/10.1037/h0025848.

[51] R.B. Zajonc. 'Mere Exposure: A Gateway to the Subliminal'. In: *Current Directions in Psychological Science* 10.6 (2001), pp. 224–228. DOI: 10.1111/1467-8721.00154. eprint: https://doi.org/10.1111/1467-8721.00154. URL: https://doi.org/10.1111/1467-8721.00154.

[52] Justin Zobel and Alistair Moffat. 'Inverted Files for Text Search Engines'. In: *ACM Comput. Surv.* 38.2 (July 2006), 6–es. ISSN: 0360-0300. DOI: 10.1145/1132956.1132959. URL: https://doi.org/10.1145/1132956.1132959.